

**KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES  
TECHNOLOGY**

Yang Xiaojie

**Analysis of DBMS: MySQL Vs PostgreSQL**

The Bachelor's Thesis Information Technology program  
Kemi 2011

## **PREFACE**

I would like to thank my supervisor Aalto Teppo in the first place. Without his patient help and guide I cannot achieve my goals in this thesis.

Secondly, I would like to thank those people how have helped and support me, included my family, friends, school staffs.

At last, I like to thank the teachers and courses that helped me to obtain knowledge and experiences during my 4 year studies in Kemi-Tornio University of Applied Sciences. It is them which inspired and helped in very phases of my thesis life-circle.

## ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology	
Degree Programme	Information Technology
Name	Yang Xiaojie
Title	Analysis of DBMS tools: MySQL Vs PostgreSQL
Type of Study	Bachelor's Thesis
Date	11 October 2011
Pages	60+7 appendices
Instructor	Aalto Teppo, M.Sc(Tech.)
Company	Kemi-Tornio University of Applied Sciences
Contact Person/Supervisor from Company	Aalto Teppo, M.Sc(Tech.)

The purpose of this thesis is to compare the functionalities of Database Management System tools of two world-famous software programs: PostgreSQL and MySQL under the consideration of both commercial views and technical aspects.

As the reason that they have conjunctly occupied a pivotal role in the field of DBMS, therefore, the comprehensive comparison of the selection criteria of these two software programs is under a great attention of varied users. According to the commercial market analyze and anatomize the specific technical functionalities of the two programs, readers could obtain a clear view of the usability of the programs while standing on a high vantage point and have a farsighted view of their future. The similar topic has not existed in the Electronic library Theseus before which means it is a terrific opportunity to make up the empty field. The DBMS concepts and factors support the thesis content through the whole exposition structure that can be regard as reading tips for readers.

The empirical research is mostly based on analysis and compare of statistical data that collect from Internet authorities and manual references, the result of individual experiments that performed on different platforms. These help to identify differences between MySQL and PostgreSQL. Communications with supervisors also plays an important role as a vital source of inspiration. However limitations are still existed as the un-accessible to those confidential statistics and the program's core trade secrecies. Limitation also emerged because that the testing equipments were too persuasive enough to make conclusions which is satisfied the scientific criteria. More examines and tests are advised if the thesis result is needed to be used in academic.

Keywords: DBMS, MySQL, PostgreSQL, important factors in DBMS, selection criteria

## TABLE OF CONTENTS

PREFACE .....	I
ABSTRACT .....	II
TABLE OF CONTENTS .....	III
EXPLANATION OF CHARACTERS AND ABBREVIATIONS .....	V
1. INTRODUCTION .....	1
2. HISTORY OF THE DATABASE MANAGEMENT SYSTEM .....	3
2.1. Relational DBMS .....	4
2.2. SQL standard .....	4
2.3. History of MySQL .....	6
2.4. History of PostgreSQL .....	9
2.5. Open source software .....	11
3. MYSQL AND POSTGRESQL MARKET EXPOSITION .....	13
3.1. Ease of use .....	13
3.1.1. MySQL Installation .....	13
3.1.2. PostgreSQL Installation .....	18
3.2. Licensing .....	21
3.2.1. MySQL .....	21
3.2.2. PostgreSQL .....	22
3.3. Features .....	22
3.3.1. MySQL .....	23
3.3.2. PostgreSQL .....	25
3.4. Support .....	26
3.5. Scalability and reliability .....	27
3.5.1. MySQL .....	28
3.5.2. PostgreSQL .....	28
3.6. Short Summary .....	28
4. ANALYSIS OF POSTGRESQL AND MYSQL .....	31
4.1. Architecture .....	31
4.2. Data model .....	33
4.2.1. Data architecture .....	34
4.2.2. Data operations .....	35
4.2.3. Data Integrity Constraints .....	36
4.3. Query and query processing .....	36
4.3.1. Support SQL syntax standard .....	36
4.3.2. Stored Procedure .....	36
4.3.3. Triggers .....	37
4.4. Advanced query and optimization .....	38
4.4.1. Subqueries .....	38
4.4.2. Indexing .....	38
4.5. Transaction and concurrency control .....	40
4.6. Data storage and partition .....	42
4.7. High availability, replication and recovery .....	44
4.7.1. MySQL .....	45

4.7.2. PostgreSQL .....	47
4.8. Encryption and authentication.....	49
4.9. Support for distribution and parallel processing .....	50
5. CONCLUSIONS .....	51
6. REFERENCES.....	52
7. LIST OF APPENDICES .....	60

## EXPLANATION OF CHARCTERS AND ABBREVIATIONS

DBMS	Database Management System
DBM	Database Management
IDS	Integrated Data Store
DBYG	Data Base Task Group
CODASYL	Conference on Data System Languages
COBOL	Common Business-Oriented Language
ACM	Association for Computing Machinery
SQL	Structured Query Language
RSI	Relational Software, Inc
OO DBMS	Object-oriented Database Management System
RDBMS	Relational Database Management System
GPL	GNU General Public License
ORDBMS	Object-relational Database Management System
ANSI	American National Standards Institute
CVS	Concurrent Versions System
FreeBSD	An open source Unix-like operating system
PC-BSD	An open source Unix-like operating system
KDE	A powerful graphical desktop environment for UNIX workstations
BSD license	Berkley Software Distribution license
LDAP	Lightweight Directory Access Protocol
SSL	Secure Sockets Layer
SSH	Secure Shell
FK	Foreign key
SMP	Symmetric Multiprocessing
API	Application Programming Interface
MVCC	Multi-version Concurrency Control
LFS	Large File Support
DRBD	Distributed Replicated Block Device
SBR	Statement Based Replication
RBR	Row Based Replication
MBR	Mixed Based Replication
PITR	Point-in-time recovery
RBAC	Role Based Access Control
GSSAPI	An industry-standard protocol for secure authentication defined in RFC 2743
SSPI	A Windows technology for secure authentication with single sign-on
RADIUS	For validate the user name/password pairs.
PAM	Pluggable Authentication Modules

## 1. INTRODUCTION

The thesis topic was chosen under the circumstance that technology developing sharply. Early days, software was generally free, and it was popular and shared among limited number of researchers and developers, who were usually eager to build and promote the development of the New World of software. Things changed along with time, computers and new technologies begin to be purchase by people as the incredible influence of its advantages and convenience. Till now, technologies are been widely accepted and used in nearly all aspects of the world developing processes. Thus, the impetus of technology that integrates with commercial world can never be under-estimated; its distinct performance is keeping consolidating its pivotal status. One of the most compact conjunctions of technology and business interest is DBMS tools as it is imperative that perfectly boost both efficiency and productivity. The promoting development of DBMS progress that draws a crucial attention of both enterprises and academics to build the great demanded situation of DBMS tools analyze while with the strong recommendation from the supervisor; the topic is given priority to be taken into my account as my thesis topic. After research, making a competition of the two most popular and disputed DBMS programs comes out from my mind with all mentioned motivations above formed fundamental cornerstone of my thesis.

Through the whole study structure, self-study plays a key role among all of the others. The materials are mostly from Internet Libraries; the electronic books and authoritative manual book are common reading materials. Statistics that quoted in the thesis are collected from official websites, online questionnaires and so on. Additionally, individual experiments occupy a certain percentage to achieve the aim of supporting the practical part of the thesis research. Different platforms are been chosen to attain the commonality and increase the effort of the research persuasion.

The content of the thesis is divided into three main chapters except the introduction and the conclusion. Firstly, chapter 2 describes the information that how database management system developed, how relational database management system emerged and how it developed. In this chapter, there is also some information about history of MySQL and PostgreSQL, information about SQL standard and open source program. Chapter 2 contains the general background of this thesis. The core research content about comparison of MySQL and PostgreSQL is divided into 2 chapters – chapter 3 and chapter 4. Chapter 3 is responsible for comparing the two programs in the view from market exposition. Users opinions and basic information of this MySQL and PostgreSQL such as usability, license, basic level feature and so on. Chapter 4 will go deeper to explore the differences between these two programs in professional levels. Comparison comes from the view of technical way. The comparison covers their architecture and main functions such as data architecture,

query, and transactions and so on.

The aim of this thesis is offer information about these two famous DBMS for people who are interested in database management technology. This thesis is designed to express the objective analysis of MySQL and PostgreSQL.



## 2. HISTORY OF THE DATABASE MANAGEMENT SYSTEM

As one of the most essential aspects of data processing is data management, the data management operated by computers indicates includes: data organization, classification, coding, storage, retrieval and maintenance of operational means and approaches. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

Before the age of DBMS, the database management has experienced the Early Manual System (Before the middle of 1950s) and File Processing System (During the 1950s and early 1960s). In the Early Manual System period, data cannot be preserved, shared nor independent. It also contains a lot of redundantly duplication as it it contained in programs. The weakness is due to the limitation of computer sciences. Situation had been improved in the later periods. In File Processing System, it can solve lots of problems in the previous system. However it performance still not perfect enough to meet people's need such as data is lack of independent and has redundantly duplication. People kept expecting better database management system. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

Since the late 1960s, the database management evolution entered the DBMS. During this period, the support of both the computer software and hardware technology, especially the maturity of disk technology gives priority to the online accessing database technology. So that the system can overcomes the drawbacks of the previous data management approaches to provide a complete and more advanced data management methodology. The basic designed concept is to resolve the problem of the data sharing of multi-users, to achieve high-level centralized data management, to equip the data with high independence, and providing different means of protective action for the data. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

The trigger of DBMS is a tape drive which can input hundreds of records per second by a computer named Univac – which is discovered by the Remington Rand Inc in 1951. With the increasing demand of data sharing, the traditional file processing system cannot meet the need of the enterprises and society. As a result, the Navigational DBMS, which include the Network Database Model and the Hierarchical Database Model, emerged in 1960s. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

The Integrated Data Store (IDS) - the earliest Network DBMS as well as the first DBMS - is discovered by Charles W. Bachman who was employed in the General Electric Company as a manager of the department of program development in 1964. Its concept is linked data to a set of network data organization however it has no concept of both “find” or “search”. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

The Hierarchical DBMS was followed by the emergence of the Network DBMS. One of the most famous typical hierarchical is the Information Management System (IMS) which was developed by the IBM company in the year 1968. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

## 2.1. Relational DBMS

After the era of the navigational DBMS, one of the most profound developments in the history of DBMS had showed up. In 1970, a well-known paper "A Relational Model of Data for Large Shared Data Banks" published in <Communication of the ACM> firstly described the idea of the Relational DBMS. This paper was written by Edgar Codd who worked at the IBM Company at that time and not satisfied with the incomplete functionalities of the Navigational DBMS model. The paper later becomes a milestone of the DBMS development. The idea that he presented is by using "table" form to store data so that can prevent the situation of wasting space. With the Relational model, the database can make up space for the database if needed as the forms are all in uniform standard and linked together with a "key" data. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

In the Navigational DBMS model, all data would be stored in a single record and unessential data would simple not stored in the record. However, in the relational DBMS, optional table would be created only if the data is provided. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

Since the foundation of the relational model in 1970, the IBM Company kept discover this topic to prove its feasibility with lots of people in well-known project "System R" thought it is too ideal to be achieved in reality. At the same time, numerous companies and academic groups took step in discovering the Relational DBMS model, like the Relational DBMS named "Ingres" worked out by Eugene Wong and Michael Stonebraker of the Berkeley. In fact, the research was based on the Codd's paper and the published information of the System R. Their product then has been commercialized by the Oracle Company, the Ingres Company and the other companies in Silicon Valley. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

## 2.2. SQL standard

SQL is the short writing of Structured Query Language, which is used in database management system. Its scope includes data insert, query, update and delete, schema creation and modification, and data access control. SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". /11/ The language was created by Ray Boyce and Don Chamberlin when they tried to present mathematical definition of the 12 rules in Codd's relational DBMS theory by simple keyword syntax and it is a highly non-procedural language that do not need to be command how to do tasks. Currently it is one of the most widely used database language in the world. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

SQL was adopted as a standard by the American National Standards Institute (ANSI) in 1986 as SQL-86 and International Organization for Standardization (ISO) in 1987. Until 1996, the National Institute of Standards and Technology (NIST) data management standards program certified SQL DBMS compliance with the SQL standard. The SQL standard has gone through a number of revisions, as shown in table 1 below: /12/ /13/ /14/

**Table. 1. SQL standard history/12/**

Year	Name	Alias	Comments
1986	SQL-86	SQL-87	First formalized by ANSI.
1989	SQL-89	FIPS 127-1	Minor revision, adopted as FIPS 127-1.
1992	SQL-92	SQL2, FIPS 127-2	Major revision (ISO 9075), Entry Level SQL-92 adopted as FIPS 127-2.
1999	SQL:1999	SQL3	Added regular expression matching, recursive queries, triggers, support for procedural and control-of-flow statements, non-scalar types, and some object-oriented features.
2003	SQL:2003	SQL 2003	Introduced XML-related features, window functions, standardized sequences, and columns with auto-generated values (including identity-columns).
2006	SQL:2006	SQL 2006	ISO/IEC 9075-14:2006 defines ways in which SQL can be used in conjunction with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database and publishing both XML and conventional SQL-data in XML form. In addition, it enables applications to integrate into their SQL code the use of XQuery, the XML Query Language published by the World Wide Web Consortium (W3C), to concurrently access ordinary SQL-data and XML documents.
2008	SQL:2008	SQL 2008	Legalizes ORDER BY outside cursor definitions. Adds INSTEAD OF triggers. Adds the TRUNCATE statement.

Interested parties may purchase SQL standards documents from ISO or ANSI. A draft of SQL 2008 is freely available as a zip archive./12/ /13/ /14/ /15/

In 1976, Multics Relational Data Store - the first commercial relational database production was discovered by the Honeywell Company. The Relational Software, Inc (RSI) delivers Oracle to the open market as the world's first commercial RDBMS and the company changed its name to "Oracle Systems Corporation" in 1982. With the dozens of years' development of the relational DBMS, there are several representative products like the Oracle, DB2 from the IBM Company, MS SQL server from the Microsoft Company and Informix, ASABASD and so on. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

After 1990s, though scientists kept trying innovation in discovering new DBMS models, such as the "object-oriented database" or the "OO DBMS" which is better performed in handling the many-to-many relationships database. It is not mature enough to get success. /1/ /2/ /3/ /4/ /5/ /6/ /7/ /8/ /9/ /10/

These current developments would not have been possible without the evolution of database management. Even with the progress of database management, there is a demonstrated need for new development as specifications and needs change. As the speeds of consumer internet connectivity increase, and as data availability and computing become more ubiquitous, databases are seeing a migration to web services. Web-based languages such as XML and PHP are used to process databases. These languages allow databases to live in "the cloud." As with products such as Google's Gmail, Microsoft's Office 2010, and Carbonite's online backup services, many services are beginning to move to web based services due to increasing internet reliability, data storage efficiency, and the lack of a need for dedicated IT staff to manage the hardware./9/

### 2.3. History of MySQL

The actual story of MySQL was occurred in an in-house database tool named UNIREG for manage database, invented by Michael Widenius (a.k.a. Monty) in 1979. The working concept of UNIREG is using a low-level connection to link to an ISAM storage which contains indexing. At that time, the Swedish company TcX was on its way to build applications based on web with the UNIREG. Because of the limitation of cost, the TcX began to looking for alternatives and finally decided to make improvement themselves. /16/ /17/ /18/ /19/ /20/ /21/

The TcX made changes based on the foundation of UNIREG and by writing an API, which took full advantages of mSQL the third-party abilities. Thus, an original mSQL user who wants to transplant to the TcX's more feature-rich database server would only have to make trivial changes to any existing code. However, the code supporting this new database was completely original TcX's originality. /16/ /17/ /18/ /19/ /20/ /21/

In May 23, 1995, it was the birth of MySQL 3.11. At the time, MySQL can satisfy the internal use requirements. Moreover, TcX decided to release MySQL under the GPL-an open source license; consequently, the early MySQL was fully free. TcX evolved into the MySQL AB Company later, which had two headquarters in both Sweden and USA. //16/ /17/ /18/ /19/ /20/ /21/

A considerable progress had been took space in MySQL version 3.23 around the year 2001. This version supports most SQL operations as well as MyISAM and InnoDB storage engine. After a few years' improvements, the most classical version of MySQL has been released – MySQL 4.1 in Oct 2004,. In the later October, another significant MySQL version, MySQL 5.0 has been released which indicates its path to the high performance database more clearly. The latest version of MySQL is MySQL 5.5. /16/ /17/ /18/ /19/ /20/ /21/

Compared with the development of MySQL program, its company has also experienced lots of changes. In January 16, 2008 MySQL AB announced that Sun Microsystems acquired it for approximately \$ 1 billion. The acquisition process was completed in February 26 of the same year. In April 20, 2009, Sun Microsystems announced the company was been acquired in \$ 9.50 per share, with the total amount of \$ 7.4 billion by the Oracle. The acquisition was finally approved by the EU in January 21, 2010. /16/ /17/ /18/ /19/ /20/ /21/ /22/

## What's MySQL.

MySQL is a RDBMS (Relational Database Management System), which works based on the relational database model, to make database performance more quickly and flexible. The the SQL standard used in MySQL commonly means the current version of the SQL Standard at any time. /18/

MySQL software has dual-license; with Open Source License means it is free for private use; with commercially licenses allows MySQL can be purchased by enterprises or people who want to embed the code into commercial applications. After all, with its rich functionalities, MySQL kept to be considering as one of the most famous and reliable DBMS which occupies a large portion of database customer market. /16/ /17/ /18/ /19/ /20/ /21/

**Table. 2. MySQL Development History /18/**

Feature	MySQL Series
Unions	4.0
Subqueries	4.1
R-trees	4.1(For the MyISAM storage engine)
Stored procedures and functions	5.0
Views	5.0
Cursors	5.0
XA transactions	5.0
Triggers	5.0 and 5.1
Event scheduler	5.1
Partitioning	5.1
Pluggable storage engine API	5.1
Plugin API	5.1
InnoDB Plugin	5.1
Row-based replication	5.1
Server log tables	5.1
Scalability and performance improvements	5.4
DTrace support	5.4
InnoDB as default storage engine	5.5
Semisynchronous replication	5.5
SIGNAL/RESIGNAL support in routines	5.5

Performance Schema	5.5
Supplementary Unicode characters	5.5

The most requested features and the MySQL versions in which they were implemented are listed as shown above in the Table.2. Subqueries and R-trees function was implementing to MySQL in the 4.1 version. Big progress in the 5.0 version was the adding triggers and views. Till the 5.1 version, partitioning and row-based replication had added to the program. In the latest 5.5 version, semi-synchronous replication was new to its replication function and the system's performance has been improved. /16/ /17/ /18/ /19/ /20/ /21/

## Price (Compare MySQL Editions)

**Table. 3. Price Compare by Editions/19/**

	MySQL Standard Edition	MySQL Enterprise Edition	MySQL Cluster Carrier Grade Edition
<b>Annual Subscription</b> <sup>2,3,4,5</sup> /1-4 Socket Server /Year	USD 2,000 <b>Buy Now</b>	USD 5,000 <b>Buy Now</b>	USD 10,000 <b>Buy Now</b>
<b>Oracle Premier Support</b> <sup>3</sup>			
24x7 Support	✓	✓	✓
Unlimited Support Incidents	✓	✓	✓
Knowledge Base	✓	✓	✓
Maintenance Releases, Bug Fixes, Patches, Updates	✓	✓	✓
MySQL Consultative Support	✓	✓	✓
<b>MySQL Features</b>			
MySQL Database	✓	✓	✓
MySQL Connectors	✓	✓	✓
MySQL Replication	✓	✓	✓
MySQL Partitioning		✓	✓
<b>MySQL Workbench SE</b> <sup>1</sup>	✓	✓	✓
Storage Engine: MyISAM	✓	✓	✓
Storage Engine: InnoDB	✓	✓	✓
Storage Engine: NDB			✓
<b>MySQL Enterprise Monitor</b> <sup>1</sup>		✓	✓
<b>MySQL Enterprise Backup</b> <sup>1</sup>		✓	✓
<b>MySQL Cluster Manager</b> <sup>1</sup>			✓
MySQL Cluster Geo-Replication			✓

The numbers marked on the table 3 are explained in APPENDIX 1. The table 3 shows the prices of different MySQL editions. MySQL Standard Edition cost the least but it has five functions missing when comparing with the most expensive edition – MySQL Cluster Carrier Grade Edition. /19/

## 2.4. History of PostgreSQL

### What is PostgreSQL?

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES Version 4.21, developed at the University of California at Berkeley Computer Science Department. POSTGRES pioneered many concepts that only became available in some commercial database systems much later./23/

‘An ORDBMS is an extension of the more traditional relational database management systems (RDBMS). An RDBMS enables users to store related pieces of data in two-dimensional data structures called tables. This data may consist of many defined types, such as integers, floating-point numbers, character strings, and timestamps. Data inserted in the table can be categorized using a grid-like system of vertical columns, and horizontal rows. The relational model was built on a strong premise of conceptual simplicity, which is arguably both its most prominent strength and weakness. The object-relational aspect of PostgreSQL adds numerous enhancements to the straight relational data model. These include support for arrays (multiple values in a single column), inheritance (child-parent relationships between tables), and functions (programmatic methods invoked by SQL statements). For the advanced developer, PostgreSQL even supports extensibility of its data types and procedural languages.’/24/ /25/

PostgreSQL is an open-source descendant of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features: /23/

- Complex queries
- Foreign keys
- Triggers
- Views
- Transactional integrity
- Multi-version concurrency control.

Also, PostgreSQL can be extended by the user in many ways, for example by adding new/23/

- Data types
- Functions
- Operators
- Aggregate functions
- Index methods
- Procedural languages.

And because of the liberal license, PostgreSQL can be used, modified, and distributed by everyone free of charge for any purpose, be it private, commercial, or academic./23/

## A Brief History of PostgreSQL

The predecessor of Postgres was INGRES which is also the predecessor of many early commercial DBMS. INGRES was invented at the University of California at Berkeley (1977-1985), Professor Michael Stonebraker formed a team to develop Postgres – an object-relational database server based on INGRES at Berkeley as well. /25/ /26/ /27/ /28/

During the life circle of Postgres, several releases published. The very first example system established in 1987 and then was shown at the 1988 ACM-SIGMOD Conference. The first Version only used by a few external users in June 1989 and the Second Version was released to public in June 1990 with the new POSTGRES system. The Third Version appeared in 1991 and added support for multiple storage managers, an improved query executor, and a rewritten rule system. As a result of frequently improvement on Postgres, the size of the external user community nearly doubled during 1993. It became increasingly both expectation and financial burden, consequently, the Berkeley POSTGRES project officially ended with Version 4.2. /25/ /26/ /27/ /28/

Two Berkeley graduate students, Jolly Chen and Andrew Yu, carried up the task to improve Postgres and subsequently added SQL capabilities to Postgres. Postgres was re-named as Postgres95 (1994.1995). /25/ /26/ /27/ /28/

Postgres95 code was completely ANSI C and trimmed in size by 25%. Many internal changes improved performance and maintainability. Postgres95 release 1.0.x ran about 30-50% faster on the Wisconsin Benchmark compared to POSTGRES, Version 4.2. Apart from bug fixes, the following were the major enhancements: /24/ /28/

- The query language PostQUEL was replaced with SQL (implemented in the server). Subqueries were not supported until PostgreSQL (see below), but they could be imitated in Postgres95 with user-defined SQL functions. Aggregate functions were re-implemented. Support for the GROUP BY query clause was also added.
- A new program (psql) was provided for interactive SQL queries, which used GNU Readline. This largely superseded the old monitor program.
- A new front-end library, libpq, supported Tcl-based clients. A sample shell, psqlsh, provided new Tcl commands to interface Tcl programs with the Postgres95 server.
- The large-object interface was overhauled. The inversion large objects were the only mechanism for storing large objects. (The inversion file system was removed.)
- The instance-level rule system was removed. Rules were still available as rewrite rules.
- A short tutorial introducing regular SQL features as well as those of Postgres95 was distributed with the source code
- GNU make (instead of BSD make) was used for the build. Also, Postgres95 could be compiled with an unpatched GCC (data alignment of doubles was fixed).



Progress kept going after the developer leave Berkeley. In the summer of 1996, a new team was formed as a call to the increasing demand of open source database server. The CVS was used to help developers to share program files. During the first six months of development, they found that a single patch might break the system and they would be unable to correct the problem. Many bug reports remained and a certain number of them is duplicates, they tried fix them one by one and make note. What was interesting is that "it was amazing to see that many bugs were fixed with just one line of C code." as Jolly Chen said. /25/ /26/ /27/ /28/

In the late 1996, the team changed the database name to PostgreSQL which presents the meaning of to honor both the Berkeley name and SQL capabilities. A promote CVS was used to distributed the code all over the world via Internet, which allowed people access to the least copies of the developing tree more easily. The development schedule was still very aggressive as the developers released new updating every three to five months. Each period consisted of two to three months of development, one month of beta testing, a major release, and a few weeks to issue sub-releases to correct serious bugs. /25/ /26/ /27/ /28/

Big steps includes transaction support, complex queries, commercial-grade SQL support, complex data types, and reliability motivated the team going farther. And with their hardworking, PostgreSQL became famous and transplant to different operating systems. PostgreSQL is unique because it supports the world's most rich data types; it is the only oen source DBMS supports full transactions, multi-version concurrency control system, data integrity checks and other unique characteristics. /25/ /26/ /27/ /28/

The latest version of PostgreSQL is PostgreSQL 9.0.3. More release details about what has happened in PostgreSQL since then can be found in link:  
<http://www.postgresql.org/docs/9.0/static/release.html>. /29/

## 2.5. Open source software

The term, open source software, often confuses people. With commercial software, a company hires programmers, develops a product, and sells it to users. With Internet communication, however, new possibilities exist. Open source software has no company. Instead, capable programmers with interest and some free time get together via the Internet and exchange ideas. Someone writes a program and puts it in a place everyone can access. Other programmers join and make changes. When the program is sufficiently functional, the developers advertise the program's availability to other Internet users. Users find bugs and missing features and report them back to the developers, who, in turn, enhance the program./28/

It sounds like an unworkable cycle, but in fact it has several advantages: /28/

- A company structure is not required, so there are no overhead and no economic restrictions.

- Program development is not limited to a hired programming staff, but taps the capabilities and experience of a large pool of Internet programmers.
- User feedback is facilitated, allowing program testing by a large number of users in a short period of time.
- Program enhancements can be rapidly distributed to users.

### **3. MYSQL AND POSTGRESQL MARKET EXPOSITION**

Since there are thousands of users of both MySQL and PostgreSQL, views about these two programs are quite varied. These views are usually according to users' own experience when work with the programs and considered the related information and background of the programs. Therefore, this section analyzes the users' opinions about the two DBMS programs and briefly introduces information about how to get started with them.

Commonly, the basic consideration of selecting of a DBMS tool is, firstly, based on users' need; then about price, reputation, and ease of usability; at last they would go detail to look at the community and services and so on.

Which one should I choose, PostgreSQL or MySQL? This is a question with no one's uncertain answer. In fact, Oracle、 Sybase、 Informix might also be wise choice except those two. However, according to the users' experiences, there seems has some evaluation tips. When go to further detail from their views, analysis flowed by the concerning order.

#### **3.1. Ease of use**

According different online surveys, almost 75% percent of participators acknowledge that MySQL is easier to handle for new learners than PostgreSQL. However, PostgreSQL can better satisfy the professional users and easily operate by new but experienced users, especially meet those requirements of academics. The services that can convenient users in the most extent are not the only assessment for DBMS, people also concern a lot about how to start with it. Installation is a case in this point. /30/ /31/

##### **3.1.1. MySQL Installation**

###### **Install to Window 7 (Windows) platforms**

Installation in Windows platforms is simple. Normally, users just need to follow the instruction in the .exe install file, and click button to make confirmation. Options like install from source code or other format code are also available to users. Here take the easiest installation way as an example.

The installation file can be downloaded freely in this site:

<http://dev.mysql.com/downloads/>. There are MySQL Community Server, MySQL Cluster, MySQL Workbench (GUI Tool), MySQL Proxy, MySQL Connectors. Each of them has

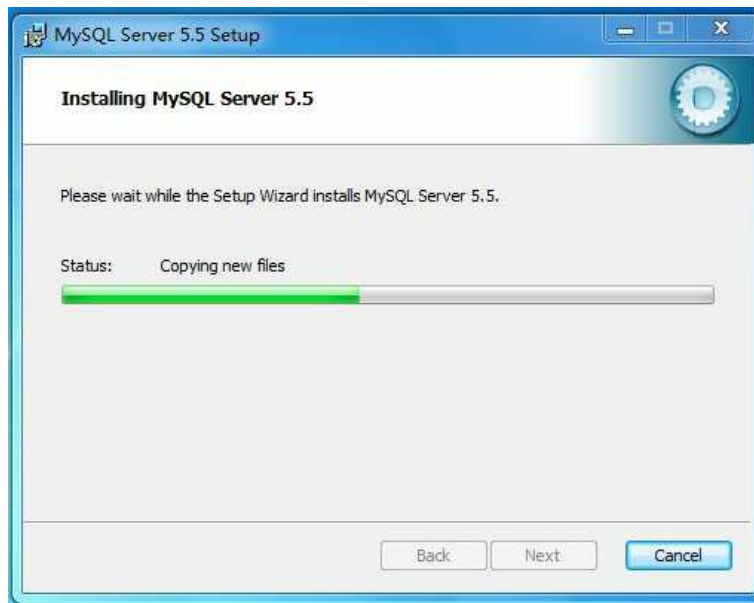
different characteristics and detail description to guide the users. The version can be chosen depending on different types and versions of platforms and installation file extensions in the download page.

The installation would be start like this, take MySQL Community Server installation as an example:



**Fig. 1. MySQL Setup begin**

- ➔ Next  
Confirm to start installation like the figure shown above.
- ➔ Next  
Confirm agree the End-User License Agreement.
- ➔ Typical -> Next  
Firstly choose setup type, then click “Next” to confirm.
- ➔ Install  
Press “Install” to start installation in the “Ready to Install” page. This process might take a few minutes as below shown.



**Fig. 2. MySQL installing**

➔ Next -> Finish

There will be a pop-up window shows up when the process is almost finished. Follow the instruction click “Next”, then “Finish” to complete the installation. Configuration takes the next step automatically.

➔ Next -> Choose “Detailed Configuration” then click “Next” in the next page.

➔ Choose “Developer Machine” then click “Next”.

➔ Choose “Multifunctional Database” then click “Next”.

➔ Maintain the default value in the “InnoDB tablespace” page, click “Next”.

➔ Choose “Manual Setting” mode to set the number of concurrent connections that you may need, and then click “Next”.

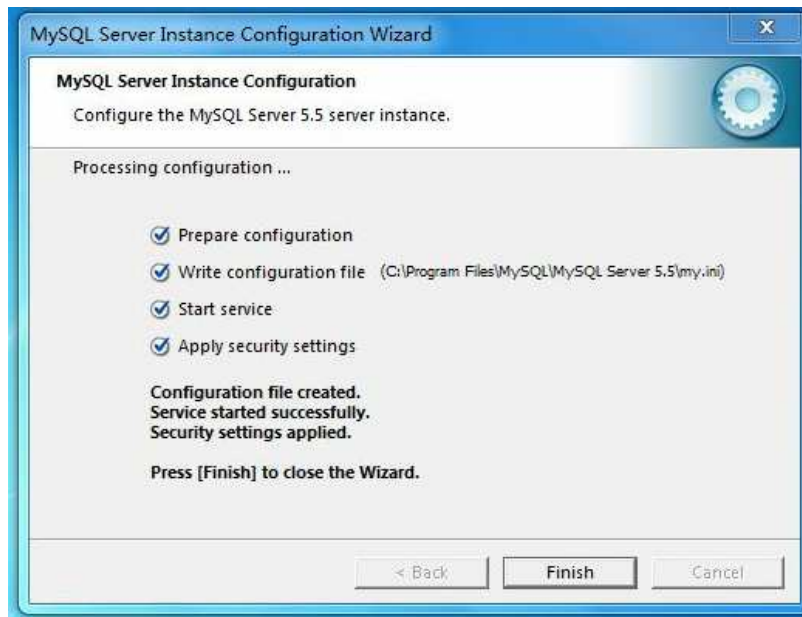
➔ Set “Network Options” and “Server SQLMode” based on need, then click “Next”.

➔ Select default character set, and then click “Next”. The “Standard Character Set” is suitable for English and European users.

➔ Set the Windows options and click “Next”.

➔ Set the security options depend on need. In this section, root password will be set and remote access can be enabled. Then click “Next”.

➔ Click “Execute” start configuration. At last click “Finish” to complete the whole installation. The finish page is shown like:



**Fig.3. MySQL finish configuration**

### **Install to Ubuntu (Linux-like platforms)**

There are several ways to install MySQL in Ubuntu, you can choose either install from source or install from package. Package can be found in Ubuntu Software Centre and start installation in there. The other way to find package is use the search function such as “aptitude search” in Terminal as super user and install the program use the install command such as “aptitude install”. The advantage of package installation is convenient because it will automatically download support program for installation. To install MySQL on Ubuntu (Linux-like platforms) from source is a way the people can control the process better and ensure the users to have the latest version of MySQL. Commanding in Terminal-window starts the installation. Firstly of all, download the source code of MySQL.

Before installation, several tools should be prepared to ensure success in execute installation no matter what kind of installation methods is. The tools mentioned in the official manual include: /32/

- CMake
- GNU make, available on system as gmake
- A working ANSI C++ compiler, such as GCC which is common used in Ubuntu
- Perl, almost included in most Unix-like systems
- Unpack tools, such as GNU gunzip; GNU tar available as gnutar, gtar, or as tar
- Bazaar in order to achieve the source tree
- Bison.

During the real installation, a package named “libncurses5-dev” might be essential. In some case, the installing process will report errors and disturbed without it. /33/

Subsequently, start to command in Terminal-Window as below: /33/

```
# Add new group and user for MySQL

shell> groupadd mysql

shell> useradd -r -g mysql mysql

# Start Building code

shell> tar zxvf mysql-VERSION.tar.gz //Uncompressed the sourcecode

shell> cd mysql-VERSION

shell> cmake .

shell> make

shell> make install

# End of building

# Initialization, offer the database ownership to MySQL

# mysql_install_db script can refresh the authorizations table

shell> cd /usr/local/mysql

shell> chown -R mysql .

shell> chgrp -R mysql .

shell> scripts/mysql_install_db --user=mysql

shell> chown -R root .

shell> chown -R mysql data

# Optional command, change configuration file location, moreover, there are 5
modes.

shell> cp support-files/my-medium.cnf /etc/my.cnf

shell> bin/mysqld_safe --user=mysql &

# Optional command to enable MySQL suto-start with systemis optional

shell> cp support-files/mysql.server /etc/init.d/mysql.server
```

### Install to PC-BSD (Unix-like) platform

The Unix-like platform that I tested in called PC-BSD. This platform designed for users who are not familiar with non-graphical system. PC-BSD is built with the kernel of FreeBSD and a graphical package handler named KDE – short for Kool Desktop Environment. /34/ /35/ /36/

The method to install MySQL in PC-BSD is called Ports service, which similar to the package installation on Ubuntu. Command in Terminal-Window is essential. The Ports service would be fully utilized in this platform as there is no need to download any pre-condition programs manually. /34/

The installation commands are: /34/

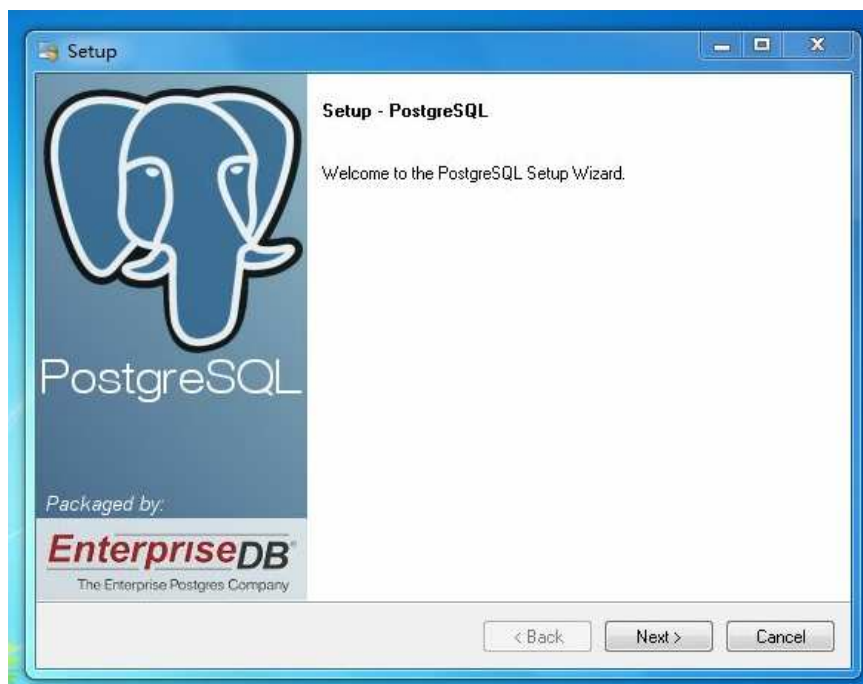
```
# cd /usr/ports/databases/mysqlversion-server/  
# make install
```

The mysqlversion represents for the MySQL version number.

### 3.1.2. PostgreSQL Installation

The installation of PostgreSQL in the same platform is similar as MySQL's. The detailed information is shown below.

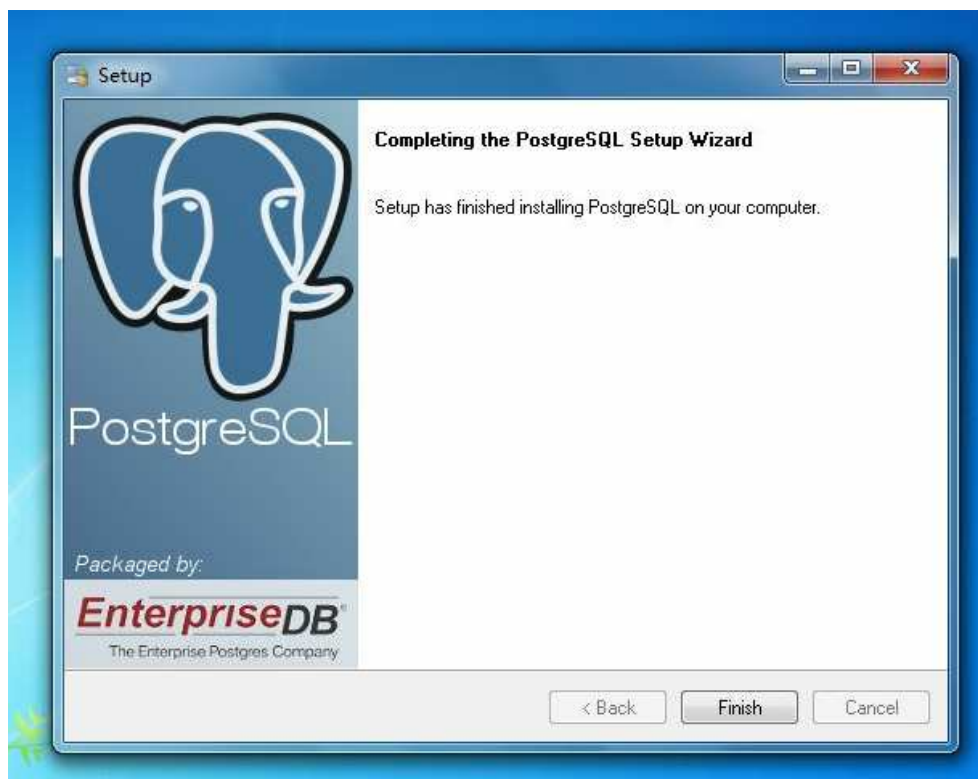
### Install to Windows 7 (Windows) platform



**Fig.4. PostgreSQL installation start page**



- ➔ Next  
Confirm to start installation like the figure shown above.
- ➔ Choose the installation directory that install PostgreSQL, and then click “Next”.
- ➔ Choose the data directory that store data, and then click “Next”.
- ➔ Setting the root password, then press “Next”.
- ➔ Select port number which is 5432 as default setting. Then click “Next” to continue.
- ➔ Point out the directory for the new database cluster, and then click “Next”.
- ➔ Click “Next” to start installation in the “Ready to Install” page. This process might take a few minutes as below shown.
- ➔ Click “Finish” to complete the installation as the figure below shown.



**Fig. 5. PostgreSQL installation finish page**

### **Install to Ubuntu (Linux-like) platforms**

The methods of install PostgreSQL on Ubuntu (Linux-like platforms) is similar as install MySQL. Package installation is preferred done with Terminal commands; the Ubuntu Software Centre doesn't offer many options for PostgreSQL. The installation from source code needs to be operated in the Terminal-Window. Firstly of all, obtain the source code file from official website. Choose the right one depend on your platform type.

Before installation, several tools should be prepared to ensure success in execute installation no matter what kind of installation methods is. The tools mentioned in the official manual include: /37/ /38/

- Make is required which is installed under the name gmake.
- Need an ISO/ANSI C compiler (at least C89-compliant).
- Unpack tools tar is required or gzip or bzip2.
- The GNU Readline library is required to help with SQL command, however it is not compulsive.

Subsequently, start to command in Terminal-Window as below: /39/

```
# Unpack the source code

gunzip postgresql-9.0.3.tar.gz

tar xf postgresql-9.0.3.tar


#Short version of installation commands

./configure

gmake

su

gmake install

adduser postgres

mkdir /usr/local/pgsql/data

chown postgres /usr/local/pgsql/data

su - postgres

/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data

/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data >logfile 2>&1 &

/usr/local/pgsql/bin/createdb test

/usr/local/pgsql/bin/psql test
```

### **Install to PC-BSD (Unix-like) platform**

The terminal commands are simple as shown below: /40/

```
# cd /usr/ports/databases/postgresqlversion-server
# make config
```

```
# make install clean
```

The postgresqlversion refers to the version number of PostgreSQL. Then the system will run installation automatically till finished. /40/

In conclusion, MySQL seems more accessible for the new users or the unprofessional users. The installation difficulty also varied from different platforms. According to my own experience, installation in Windows platform is quite easy, however, installing programs in both Linux-like (i.e. Ubuntu and so on) and Unix-like (FreeBSD, PcBSD and so on) is not nice experience in the reason that the installation requires operated in the Terminal with commands but not simple clicks.

## 3.2. Licensing

Both of MySQL and PostgreSQL are open source software. However they do under different open source license which offers them discrepant policies for different usability.

### 3.2.1. MySQL

The MySQL is available under the GPL (General Public License) license, which in order to ensure the code ability of open source, free to use and reference, modification as well as derivative work or freely adapted and distributed by all those who use the licensed software. The key point of GPL is open source code, it is not applied the code to be used a commercial software distribution and sales of modifications and derived closed-source code since the behaviors except copy, distribution and modification are not applied inside the GPL's scope. If a main content of a program include the open source code of a stated GPL product, then the program strictly required to also use GPL software protocol to be open source and free. This is the so-called "infections". The product with GPL agreement can be use and execute without any limitation as a separate product, but can enjoy the advantage of free in the same time. The freedom offered by the license benefits the extension of MySQL in database related field and enhance its popularity. /41/ /42/

Moreover, Oracle Corporation Enterprise who is publisher of releasing MySQL offered for dual-licensing for business users who do not want to work in kind of limitation and has the totally control of their code and products. Since it is a closed-license for developers and enterprises, it also has the marked price. The varied price is due to the different terms of commercial licenses like OEM licensing and so on. /41/ /42/

### 3.2.2. PostgreSQL

The license for PostgreSQL is much simpler; it has only one license called BSD license, which is also an open source license. The license is similar to the GPL that allows individuals or enterprises to distribute, copy, modify, derivate the source code. This agreement relatively offers more freedom to the users in that the users can almost "do whatever they want"; the users are even allowed to release their modified or derivative work as proprietary commercial product to earn profits or as open source which is essentially different from the GPL agreement in this point. This is extremely fit the need of those individuals and enterprises who wish to use a free and open source DBMS but develop their own closed code. /42/ /43/

While doing the secondary development of the open source code that under the BSD agreement, following requirements should be achieved: /42/ /43/

- If the re-released version product contains the source code, the original open code must with a BSD agreement in the source code.
- If only publishing the binary library or software program, the original BSD agreement must be included in the libraries or software documentation and copyright notices in the source code.
- Must not use the reputation of the open source author, institution name or original name of the product to do any forms of promotion.

Owing to the characteristics of the BSD license, a modified and derivative product with a closed source license needn't to be offer BSD license by PostgreSQL. As a consequence, on the one hand it has the strength for users that they can write closed source software without paying a fee; on the other hand it also has the weakness that not needing the users as the DBMS author to connect with the code in the programming ethics. /42/ /43/

The spirit of BSD license is to encourage sharing code with the respect to the code of the copyright. It is a friendly business integration agreement so that a vast number of companies and enterprises preferred open source products with BSD protocols when selecting, thereby completely controlling these third party codes, and modifying or secondary develops them if needed. /42/ /43/

### 3.3. Features

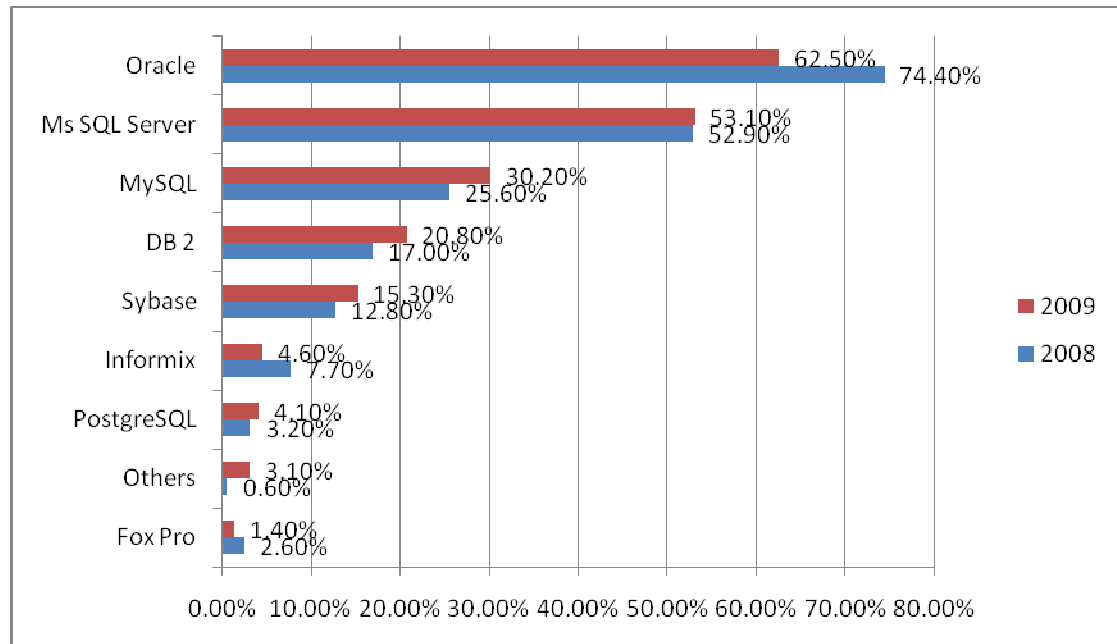
Definitely, different DBMS has different features and characteristics. Even though they might have similar functionalities, they do have dissimilar ways to be distinguished from the others. In general, these two DBMSs are robust, open source RDBMSs.

### 3.3.1. MySQL

First of all, the processing speed of MySQL is announced that one of MySQL's developers' main pursuing ideal objectives is speed in early years. Due to this goal, MySQL announced in their earlier documentation that they did not prepare to support the transaction and triggers functionalities. However, later modification was made in MySQL 4.0.2-alpha – it has begun to support transactions. Implementing more functions definitely slowed down the process speed of MySQL. And currently MySQL's speed is only faster than PostgreSQL's in several cases. Anyway, speed is one of MySQL's features that many users talking about. /44/ /45/ /46/ /47/ /48/ /49/

Secondly, MySQL is much more popular than PostgreSQL. A commercial product cannot survive without popularity as a fundamental indicator. More popular has meant more users, then means more withstood the test and better business support as well as more comprehensive and detailed documentation. /44/ /45/ /46/ /47/ /48/ /49/

Statistics in figure 6 below shows a research result of a multi-choice questionnaire about the main DBMS support distribution of business system recent years in China.



**Fig. 61. DBMS distribution 2008-2009 in China/31/**

Among the users in the survey as figure 6 shown, 62.5% of them were using the Oracle database to support their business systems. The ratio was declined nearly 12% than in 2008, This might due to the influence of the financial crisis, many companies and enterprises chose database products while concerning more about the cost. Thus, the increasing

utilization rate of MySQL, accounting for 30.2% of the total users, can be illustrated as a result for that concern. However, the users of PostgreSQL are only around 4.1% with a slight rise of 0.9% in 2009. /31/

Additionally, MySQL is more suitable for running on the Windows environment. Because they considered that Windows platform is the most mainstream platform which owns largest customer group, they began to design MySQL for the Windows platform users for a long time. When MySQL runs on Windows platform, it works as a Windows local application program. PostgreSQL, on the other hand, transplants to the Windows platform later after the 8.0 version in recently years. And PostgreSQL runs in Cygwin environment, Cygwin is a Linux-like environment for Windows making it possible to port software running on POSIX systems (such as Linux, BSD, and Unix systems) to Windows. Though no evidence shows that PostgreSQL cannot maintain the same stability as MySQL's on Windows platform, the obvious distinction causes uncertain attitude of the Windows users about PostgreSQL performance on Windows platforms. /48//50/

What's more, the ways that two programs used from prevent background buffering data and unnecessary old version data expand too large is different, and the way that MySQL use is more convenient than PostgreSQL. For instance, in InnoDB storage engine of MySQL, it offers a purge function to help clear the background unnecessary data. 'InnoDB is a high-reliability and high-performance storage engine for MySQL. Starting with MySQL 5.5, it is the default MySQL storage engine.' In InnoDB, deleted data, old data and deleted records are all stored in a place named the rollback segment. The purge action can clean up both the deleted data from both table itself and the rollback segment. It is easy to delete the unnecessary data and recover or find back data from the rollback segment. Purge keeps working in the background of MySQL to service InnoDB better because it to make ensures the database running normally and limit the database size from growing too large. While in the case of PostgreSQL, it uses VACUUM service. VACUUM runs every short period. 'Lacking a centralized record of what must be purged; PostgreSQL's VACUUM has historically needed to scan the entire table to look for records that might require cleanup. Beginning in PostgreSQL 8.3, there is an optimization called HOT (for "heap only tuple") which allows some vacuuming to be done on the fly in single-page increments; beginning in PostgreSQL 8.4 and higher, the system maintains a bitmap, called the visibility map, which indicates which pages of the table might possibly contain tuples in need of cleanup, and VACUUM can scan only those pages. However, a full scan of each index is still required during each VACUUM; make it still a somewhat expensive operation for large tables.' For users' convenience, MySQL's Purge function service better than VACUUM in PostgreSQL. /51/ /52/

Furthermore, MySQL offers the user the ability to select suitable storage engines they preferred to use, but PostgreSQL doesn't offer such kind of service. For example, in MySQL MyISAM is good if a vast number of SELECT or UPDATE commands need to applied. MyISAM is the default engine of MySQL before version 5.5 and it is also a storage engine which support transaction, index and full-text search. InnoDB can be chosen if a lot of INSERT commands need to operated because it runs better than MyISAM when with INSERT. This user-defined function offers the users convenience in some extent because the users are able to choose suitable engines. Currently the default engine of MySQL is InnoDB. /53/ /54/

Another great feature of MySQL is that it can run in embedded devices and similar low-memory condition. Though support transaction might slow down the processing speed, it is not a disadvantage of MySQL. Because storage engines which with or without transaction functions are all provided by MySQL. Because of options are offered for users. This becomes another advantage of MySQL on the other hand. /44/ /45/ /46/ /47/ /48/ /49/

### 3.3.2. PostgreSQL

Through decades of development, PostgreSQL has known as the market's most advanced open source DBMS in database market. As a full-featured open source relational database management system, PostgreSQL offers characteristic services that support high-transaction, mission-critical applications. The transaction feature, compared with MySQL, experienced more thorough tests, which is essential for solemn commercial applications. /44/ /45/ /46/ /47/ /48/ /49/

One of the core competitiveness of PostgreSQL is the security protection of data through the use of enterprise authentication mechanism that provides high-quality secure safeguard. The Lightweight Directory Access Protocol (LDAP) Assertion Control is a case in point; once action passed the validation, all communication access to the database can followed the SSL connection that provides a high-level of security protection. This part will be detailed explained in Chapter 4. /44/ /45/ /46/ /47/ /48/ /49/ /55/

While adding or modifying data, PostgreSQL enforces a large number of customized constraints, to ensure data quality qualifies the limitation of business rules, including the examinations from simple scope to complex foreign key (FK) check. As soon as the data stored to storage, it can be backed up. What's more, this is crucial to recover from crash and accidents. The support of constraint is profound guarantee the data integrity assurance; rationally utilize this service will surely lighten the work burden. /44/ /45/ /46/ /47/ /48/ /49/

Allowed the behavior of add-on modules to its core architecture by other communities or groups, PostgreSQL can create more advanced features. Consider the geographical space support of PostgreSQL for example; it is based on a module called PostGIS. It is simple module expansion like this which made PostgreSQL a more powerful DBMS with the feature of storing the spatial data. The PostGIS adds support for geographic objects to the PostgreSQL object-relational database. /44/ /45/ /46/ /47/ /48/ /56/

Another special extension ability of PostgreSQL is its stored procedural language is not limited but support multiple different types of languages, allowing developers to use languages they are familiar with to coding server-side, for instance, a trigger that need to perform complex text processing can be written by Perl language to take advantage of its powerful regular expression function. The language that it supports also included Python, Java, and C++ and so on. Additionally, it is easily connected by numerous languages like C++, Java, Ruby, PHP and many others. Moreover, the UDF (user-defined functions) can be conveniently used to extend PostgreSQL. /44/ /45/ /46/ /47/ /48/ /57/

Here is a clear conclusion of the elementary information of MySQL and PostgreSQL in table form:

**Table. 4. Basic comparison of MySQL and PostgreSQL /41/ /43 /46/ /48/**

DBMS	MySQL	PostgreSQL
Maintainer	Oracle Corporation	PostgreSQL Global Development Group
Software License	GNU General Public License (GPL) and a variety of commercially licenses	Berkley Software Distribution (BSD) license.
Max Database Size	Theoretical Unlimited(256TB)	Unlimited
Max Table Size	MySAM: 256 TB; InnoDB: 64 TB	32 TB
Type System	Static	Static
Support Windows	Yes, native on Microsoft Windows	Yes, can run on Microsoft Windows since version 8.0.
Support Linux	Yes, in most Linux distributions	Yes, in most Linux distributions
Support Unix	Yes, runs on many Unix-like operating system	Yes, runs on many Unix-like operating system
Useable	Small to medium sized database	Medium sized database

As table 4 shows, the most elementary features of MySQL and PostgreSQL are similar. However, as the analysis shown before, MySQL has better processing speed than PostgreSQL and they have different Maximum table size though they both support unlimited maximum database size. /41/ /43 /46/ /48/

Pursuing speed is one of the main goals of MySQL. However, according to the tests results, PostgreSQL storage engine performed better than InnoDB's performance. Tests also covered another storage engine of MySQL- MyISAM. MyISAM was the former default storage engine of MySQL but not comparable with PostgreSQL storage engine and InnoDB. PostgreSQL storage engine and InnoDB are comparable because they offer the same functions such as transaction, MVCC (Multi-version Concurrency Control) and so on. Information of these functions will explain in chapter 4. Detailed tests information about test environment and result are explained in APPENDIX 2.

### 3.4. Support

There is another big difference between MySQL and PostgreSQL, which is the community.



The open source project, essentially, freshened from the participation and contribution of community developers. There are two types of open source communities. /44/ /45/ /46/ /47/ /48/

First one is the pure open source project with independent community. PostgreSQL belongs to this type of communities, which is the oldest and largest independent open source database community. The strength of this kind of community is that it is definitely independent from commercial circles and won't be controlled by any commercial action. /44/ /45/ /46/ /47/ /48/

The second type of open-source community is controlled by business enterprises, which are often offers free versions and the commercial editions at the same time. MySQL is on the other side of PostgreSQL, represent as an example of this second type of open source community. MySQL project was originally financed and controlled by the MySQL AB; all the core developers and architects are hired by MySQL AB. After the acquisition by Sun, MySQL community was inherited under Sun's control, now controlled by Oracle. /44/ /45/ /46/ /47/ /48/

No matter what types of community they are, as the widely used of MySQL, the community keeps up with the customer size to assist and offer advices for the problems that might encounter in frequently. On the contrary, PostgreSQL does have a host of mailing lists and forums, in spite of there are only a few commercial concerns offering support services. /44/ /45/ /46/ /47/ /48/

### **3.5. Scalability and reliability**

Scalability of a DBMS did not under a very solemn consideration when DBMS was packaged with small applications in early years. The situation changes due to an increasing number of enterprises began to rely on open source database management systems under current economics. When open source DBMS has been more widely being accepted by more corporations, it is fundamental that those open source solutions can behave scalability. /44/ /45/ /46/ /47/ /48/

Reliability is another essential feather that the users need. Usually, reliability clasps scalability for the reason that a DBMS with weak reliability is not trustable and useful even if it has good scalability. When running in the situations such as with heavy loads or crash suddenly happened, DBMS do need good reliability to recover quickly and stable from problems. Therefore scalability and reliability combined together and play a pivotal role as guidelines to be taken into account when choose a DBMS. /44/ /45/ /46/ /47/ /48/

### 3.5.1. MySQL

Since the servers take more CPUs into consideration but not faster CPUs, MySQL fully utilize the functionality of symmetric multiprocessing (SMP) to assign loads more logically thereby managing loads more efficiently. This optimization ensures that the running process is independent no matter which kind of processor MySQL is running on. Because of MySQL focus on both software and hardware tendency to meet the future need, its scalability and reliability keep improving dramatically. /44/ /45/ /46/ /47/ /48/ /58/

Additionally, MySQL also keeps an eye on human oriented design to convenient the user work more efficiently. Take the query analyzer for example, it helps to show similar query commands and solve query problems. Despite the fact that this service is not free and available in the enterprise edition of MySQL, it does improve scalability for MySQL. /44/ /45/ /46/ /47/ /48/

### 3.5.2. PostgreSQL

PostgreSQL, identical to MySQL, tries performance reliable and scalable with its numerous services. The services included of make convenience for recovery data like point-in-time recovery; of work more efficiently like asynchronous replication and online backups and so on. PostgreSQL is always the pioneer in DBMS development field; and acquire multiple services ahead of those closed-source DBMS and expensive and proprietary DBMS. The support of these services forms the system as a most robust one and also makes contribution to increase its scalability and reliability. /44/ /45/ /46/ /47/ /48/

## 3.6. Short Summary

In fact, the favor of choice is more decide by personal taste and need but not that much depend on others' recommendation. People would try both to make a fair decision. Debate about which one is better always takes place in database forums and never ends. There might has one said: "Mysql is an answer if you have to choose one among them. While Postgresql does offering a great and flexible but Mysql is absolute free and more secure", another one thinks that: "I don't agree, MySQL might be free in most cases, it's not in every case. PostgreSQL is always for free. The BSD-licence gives you more freedom. And security, PostgreSQL is considered safer as MySQL, just check The Database Hacker's Handbook: "By default, PostgreSQL is probably the most security-aware database available ..."PostgreSQL has more and better options to secure the database." /44/ /45/ /46/ /47/ /48/

There is no clear winner according to this context. New users keep coming in and should

draw their own conclusions based upon which one meet their unique needs. Both of PostgreSQL and MySQL are skillful, high-quality open source DBMSs. In reality, choosing an absolutely dominant open source DBMS is mission impossible. Either PostgreSQL or MySQL cannot be claimed that which one is better than the other one. In fact, there isn't a best but the most suitable one. /44/ /45/ /46/ /47/ /48/

PostgreSQL and MySQL have been widely used in a multitude of applications, for instance, PostgreSQL is famous for strongly support transaction-intensive enterprise applications, many sites also use it to support web applications. On the contrary, MySQL Web always is the priority choice DBMS of web applications. In a word, MySQL is quite popular in real world applications; PostgreSQL is more popular applied in academic environment. /44/ /45/ /46/ /47/ /48/

Although PostgreSQL and MySQL are categorized in a same kind as open source DBMSs, their sameness may be limited in this overlap. They do have their own characteristics differ from each other and those expensive proprietary database products. They prove themselves are worthy competitors in open source DBMS market or even in the whole DBMS market because the current economic situation. Right now selecting a suitable open source DBMS may be a good way to save a fabulous outlay. Their rich features and outstanding licensing guarantee their head positions and strong competitive to against the other DBMS products. /44/ /45/ /46/ /47/ /48/

Conclude a result from previous comparisons. Some help tips for uses to choose a product only between PostgreSQL and MySQL can list as below. Users might prefer MySQL in following situations /44/ /45/ /46/ /47/ /48/:

- If you are a regular user of Windows platform, then for your own convenience, it is better to choose MySQL. Because it is easy to install and operate in Windows platform.
- If you are not professional to DBMS, even has no idea about what the database transaction and the storage procedure are, you would choose MySQL because it is a good tutorial tool for new database learners and easier to be start with.
- If you are not require the highly integrality and solemnness of data but pursue the speed of query processing such as building forums or communities, you might choose MySQL. Because the free version of MySQL didn't offer perfect functionalities in data integrality and solemnness.

On the other hand, users would consider PostgreSQL when they are in the other status like /44/ /45/ /46/ /47/ /48/:

- If you are doing a salmon business application in highly-level demand of data integrality. Additionally, you need excellent encapsulation of some commercial data logic, for instance for a net bank, PostgreSQL should be a better choice. Because PostgreSQL is designed for high data integrality and solemn uses.
- If you are dealing the geographic data, because of the R-trees indexing extension, PostgreSQL can service better.

- If you are a Fanatical fancier of database who wish has a database in your own version with your own discovery of DBMS, undoubtedly, PostgreSQL is a great choice because you can add, delete or modify its source code for your won need.

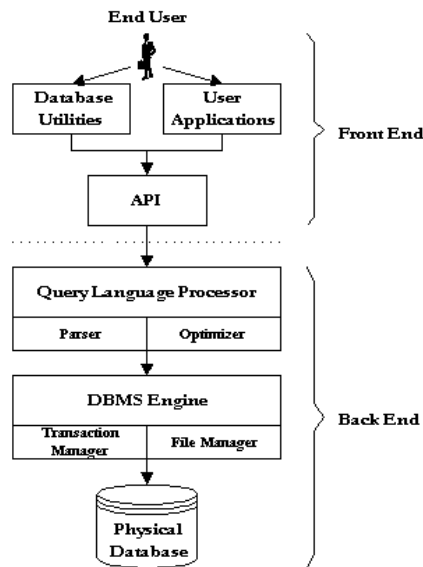
## 4. ANALYSIS OF POSTGRESQL AND MYSQL

PostgreSQL and MySQL are two database management system programs. A Database Management System (DBMS) is a set of specific programs that establish and manage database which is neither an application program nor an operating system. It contains the basic functionality to manage database such as define, build, maintain and control the database. Moreover, it can ensure database's integrity, security, concurrency control of multi-users, as well the system recovery when failures or crashes occur on the DBMS. Though they are DBMSs, the ways that they used to do a same thing are different as well as the functions that they offered for users./44/ /45/ /46/ /47/ /48/ /59/

### 4.1. Architecture

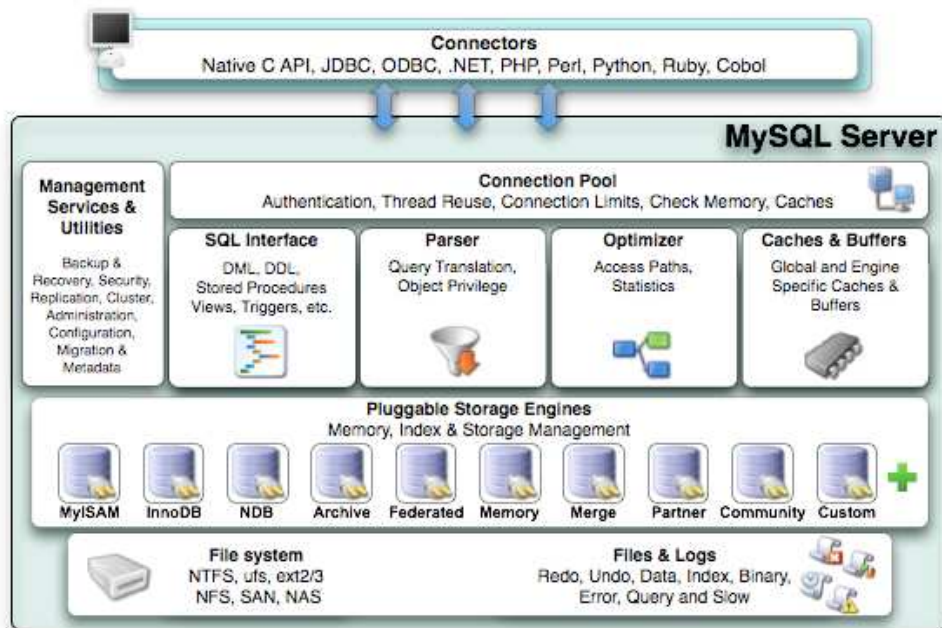
When looking deep into the common DBMSs' architecture, there are two parts: logical and physical architecture. The logical DBMS architecture manages approaches to store and present data to the users. The physical architecture focuses more on the software building blocks to form the system. /44/ /45/ /46/ /47/ /48/ /60/

Because the logical architecture describes the abstract levels of DBMS to just shows how it looks and no distinction between different DBMSs. Therefore technical comparison of DBMSs usually focuses on the physical architecture. Analysis base on physical architecture can clearly illustrate the efforts of different components that help system working efficient. /44/ /45/ /46/ /47/ /48/ /60/



**Fig.7. Physical DBMS Architecture/60/**

The figure 7 above shows the physical architecture of a DBMS. Generally, the end users would use the Application Programming Interface (API) connect to the database with different languages. Then data would process to the Back End with the flow from Query Language Processor to DBMS Engine, subsequently to the Physical Database. However, the end users who work with the front end usually do not aware of the back end which services for them in the background. /60/



**Fig.8. MySQL Architecture with Pluggable Storage Engines/61/**

MySQL has 3 layers in its physical architecture as figure 8 shown. In the first layer, a variety of languages can be used as APIs in different languages such as C, JDBC, ODBC, .Net, PHP, Perl and so on. The second layer which contains the core elements is more complex than the previous layer, because the second layer is combined by numerous parts. Data will be analyzed and optimized in this phase then go across to the third layer: Storage Engines, then finally reach the Physical Database. The storage engines have different characteristics and can be chosen when creating database based on users' need. For instance, InnoDB is commonly used for the purpose of high-performance with a vast number of concurrency; MyISAM is used for dealing with situations which have low concurrency workload or high concurrency with read most. It is also possible to combine engines together to take fully use of their varied advantages. The function that users can choose desirable storage engines in MySQL is considered as one of MySQL's most flexible feature. /61/

However, the physical architecture of PostgreSQL is simpler than MySQL's. The architecture of PostgreSQL also consists of 3 layers. In the first layer, PostgreSQL supports different languages as APIs like MySQL. In second layer, PostgreSQL also has the facilities to analyzed and optimized SQL statements. Distinction between PostgreSQL and MySQL emerges in the third layer. MySQL has several pluggable storage engines, but PostgreSQL only has a single storage engine. PostgreSQL contains an integrated database server with a single storage engine. Though the structure of the physical architecture of MySQL and PostgreSQL are all divided into 3 layers, architecture of MySQL and PostgreSQL's are different. /23/

## 4.2. Data model

A data model is an abstract, self-contained, logical definition of the objects, operators, and so forth, that together constitute the abstract machine with which users interact (the objects allow us to model the structure of data, the operators allow us to model its behavior). /3/

The data model consists of three essential components: data architecture, data operation and data integrity constraints. Data architecture can also regard as a set of data types, which describes the static features of system; while data operation represents a set of operations that allows to works on data entities as well as operation rules, which defines dynamic features of system. Moreover, data integrity constraints are a set of data regulations that enable the exactitude, integrity as well as compatibility of data. /3/

### 4.2.1. Data architecture

#### Data types

PostgreSQL supports a vast set of native data types. There are 8 main categories available to users such as Numeric Types, Monetary Types, and Character Types and so on. Especially the support of Geometric Types impresses the users a lot. Moreover, it is also possible to add new types to PostgreSQL using the CREATE TYPE command. /62/

However, though MySQL does support numerous data types, it is not competitor which is strong enough when comparing with PostgreSQL. The reason is that some of the native data types in PostgreSQL are not supported by MySQL. In order to make up this disadvantage, MySQL provides the functionality that allows users import data types from other database systems into MySQL; however the number of the imported types is still limited. /63/

#### Default Values

In PostgreSQL, a column is able to be assigned a default value. In another situation if there is a new row is inserted to a table without fulfilling the blanks, the empty columns will be filled with default values respectively according to the data types. Or if no default value is claimed clearly, the default value is be set as null value. The null value is used for representing unknown data. /62/ /64/

PostgreSQL has the functionality that when using a data manipulation command, default values will be set in empty field even if the value is unknown. Additionally PostgreSQL allows setting default value for a whole column when the function is marked as either IMMUTABLE or STABLE. /62/ /64/

However, MySQL has more complex rules about the default values. In MySQL, the DEFAULT value clause is needed in order to declare the specify default value for a column. Functions like NOW() and CURRENT\_DATE cannot be set user-defined default values. Nevertheless users can expect to specify CURRENT\_TIMESTAMP in order to set the default for a TIMESTAMP column. /63/

Additionally, BLOB and TEXT cannot set default values. If there is no clearly definition of the default value, MySQL will indicate the column NULL as default value with a DEFAULT NULL clause, if the column cannot use NULL as a value, MySQL will define the column does not have a DEFAULT clause. Situation varies if the part is defined as PRIMARY KEY. /63/

Implicit defaults data types are defined as follows/63//65/:

- For numeric types, the default is 0, with the exception that for integer or floating-point types declared with the AUTO\_INCREMENT attribute, the default is the next value in the sequence.



- For date and time types other than `TIMESTAMP`, the default is the appropriate “zero” value for the type. For the first `TIMESTAMP` column in a table, the default value is the current date and time.
- For string types other than `ENUM`, the default value is the empty string. For `ENUM`, the default is the first enumeration value.
- `SERIAL DEFAULT VALUE` in the definition of an integer column is an alias for `NOT NULL AUTO_INCREMENT UNIQUE`.

#### 4.2.2. Data operations

The data operations of both MySQL and PostgreSQL are mainly controlled by SQL commands. However, there exist numerous differences. Basically, PostgreSQL follows the SQL standard strictly but MySQL does not. A few examples will indicate the discrepancies between these two programs.

MySQL has created some functions that transform from the original SQL standard. Take the 'INSERT IGNORE' and 'REPLACE' statements as examples. These two statements are used to cover the old rows with the new rows which have the same unique key values of the old rows. The row will be replaced but not aborted. When INSERT IGNORE happens, the `AUTO_INCREMENT` counter will not increase and `LAST_INSERT_ID ()` will return 0. `REPLACE` is a statement which transforms from the original SQL standard. Usually `REPLACE` can be regarded as `INSERT IGNORE`, there is only one exception. If there is an old row in the table which contains same value of `PRIMARY KEY` or a `UNIQUE` index as a new row, then new row can be inserted after the old duplicate row is deleted. However, currently PostgreSQL doesn't support these unoriginal SQL statements and users have to use composite standard SQL statement to achieve such functions which is not strictly follow the SQL standard. /66/ /67/

A similar statement `INSERT ... ON DUPLICATE UPDATE` of MySQL is an extension to the SQL standard and does not exist in PostgreSQL because the statement does not follow the original SQL standard. Even though those derivative commands of MySQL might convenient a certain number of users, it does not mean that the action of not following the original SQL standard as a DBMS is worth encouraging. /66/ /67/

In order to convenient users but keep following the SQL standard strictly, PostgreSQL global group work harder. Right now PostgreSQL builds new future plan to add new features to service users more convenience. Such as a `MERGE` clause which follows the SQL 2008 standard. /67/ /68/

### 4.2.3. Data Integrity Constraints

Both PostgreSQL and MySQL support integrity constraints including Not-Null, Unique, and Primary Key. Foreign Key constraint is supported by PostgreSQL for a long time, however only MySQL's InnoDB storage engine support this Foreign Key constraint. /64/ /69/

MySQL begins to support ENUM and SET Constraints on the entry of invalid data; they were available as true constraints on entry of invalid data before. ENUM and SET Constraints are not the constraints that follow the SQL standard, so they cannot work in MySQL's strict SQL mode. PostgreSQL supports all MySQL's constraints which follow the SQL standard; additionally PostgreSQL supports constraints which MySQL does not have such as CHECK constraints and Exclusion Constraints. The Exclusion Constraints are used to helping compared on the specified columns or expressions using the specified operators. /69/

## 4.3. Query and query processing

### 4.3.1. Support SQL syntax standard

MySQL supports part of SQL syntax which belongs to ANSI SQL standard. Typically MySQL is running on its special SQL mode. However it is possible to switch to running on strict SQL mode - ANSI Mode. Moreover there are extensions to standard SQL in MySQL. Though MySQL and PostgreSQL seem to support same functions, but how they work are not exactly the same. For instance, there are differences between SELECT INTO TABLE and UPDATE SQL syntax though they are related extensions. MySQL doesn't follow the SQL standard strictly, but it offers specific SQL syntax in order to convenient users and these specific SQL syntaxes are considered as a feature of MySQL. On the other hand, PostgreSQL follows the SQL standard strictly and tries to support all the DBMS functions that have ever existed, such as WITH Queries (Common Table Expressions) as well as window function which is not supported by MySQL. /70/ /71/

### 4.3.2. Stored Procedure

MySQL and PostgreSQL support stored procedure in different ways. MySQL supports a stored routine to store a procedure or a function. The statement used to create a procedure or a function in MySQL is CREATE PROCEDURE and CREATE FUNCTION. The ways that PostgreSQL supports stored procedure service are using functions and triggers.

In PostgreSQL functions can be stored by the statement CREATE FUNCTION. And the catalog pg\_proc stores information about functions. Trigger also plays an important role in query optimization and will be explained later./72/ /73/ /74/

PostgreSQL uses the professional procedural language - PL/pgSQL as the first recommend choice, similarly to Oracle's PL/SQL. PostgreSQL also supports SQL:2003 PSM stored procedures as well as many other loadable procedural languages such as Perl (PL/Perl), Python (PL/Python), TCL (PL/Tcl), Java (PL/Java) and C (PL/C). PL is short for Procedural language. PSM is short for Persistent Stored Modules and it is an extension to SQL. /72/ /73/ /74/

Compared with PostgreSQL, MySQL follows the SQL:2003 syntax for stored routines, which is also used by IBM's DB2. /20/ However, with the plug-in interface the range of support language expanded, Java, Perl, and XML-RPC etc. can be used in external language stored procedures. /72/ /73/ /74/

### 4.3.3. Triggers

There is a definition of trigger according to MySQL 5.5 Reference Manual. A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. /72/ /75/ /76/ /77/

Both MySQL and PostgreSQL support triggers but with some differences. Triggers in PostgreSQL can be executed by any user-defined functions from one of the procedural languages but not only PL/pgsql. Only the SQL statements are able to activate MySQL triggers. Triggers of MySQL cannot be activated by changing in views or changing in tables, which is made by APIs, because the APIs will not transfer the SQL syntax to MySQL server. In MySQL the CREATE TRIGGER or DROP TRIGGER statements are used to create or drop triggers. /72/ /75/ /76/ /77/

PostgreSQL supports triggers as well as the rules system which can also achieve to perform the triggers' functions. However, some types of constraints cannot apply by rules, system especially the foreign key. Additionally, a trigger in PostgreSQL cannot be created in view since there isn't any real data entity. The rules system, can also named as the query rewrite rule system is quite differ from stored procedures or triggers. It is powerful and can be applied-widely to a host of different aspects, such as query language procedures, views, and versions. The work process of it can be conclude in this way, firstly it will take rules into account after modifying queries, then transfer the after-processed queries to the query planner, the queries later will be planned and executed. /72/ /75/ /76/ /77/

Similar as MySQL, PostgreSQL uses CREATE TRIGGER and DROP TRIGGER to create and remove triggers. Currently, PostgreSQL provides an inner trigger function: suppress\_redundant\_updates\_trigger. It helps to prevent the situation such as updating for a row which doesn't truly change data. /72/ /75/ /76/ /77/

To define a trigger, the syntax is not that forthright in MySQL because the syntax needs make definitions separately for functions with particular data type returned. In the reason of this, PostgreSQL performance better in the trigger service. /72/ /75/ /76/ /77/

#### **4.4. Advanced query and optimization**

Advanced query and index are significant methods to optimize query. Usually, advanced query includes the actions to search information with different table or column combination in desired order. This can be achieving by JOIN function which is supported by both MySQL and PostgreSQL. More details about query optimization are analyzed in subqueries and index as below shown.

##### **4.4.1. Subqueries**

Both MySQL and PostgreSQL support subqueries; MySQL started the support subqueries since the 4.1 version. The subquery types which are supported by both MySQL and PostgreSQL are EXISTS, IN, NOT IN, ANY, SOME, as well as ALL. PostgreSQL has Row-wise Comparison; however MySQL supports Row Subqueries which have the similar functions. PostgreSQL also supports WITH subquery. /78/ /79/ /80/ /81/

Because of the reason that the support of subqueries started since MySQL 4.1 which is far later than the time that PostgreSQL began subquery function; the subquery performance of MySQL is not good as PostgreSQL's and is under the situation that needs to be improved and optimized. However the improvement does not take real action even till the nearest version of MySQL. On the other hand, PostgreSQL has already taken actions to improve and optimize certain types of subquery. Even though PostgreSQL still did not allow executing subquery in view, this function can be implemented by PostgreSQL's third-party package. Executing subquery in view is a removed function from MySQL 4.1 and still not existed till now. All of the subquery expression formats in PostgreSQL are return in the format of Boolean (true/false) results. /78/ /79/ /80/ /81/

##### **4.4.2. Indexing**

In order to achieve greater performance of SQL query statement, indexes help to increase the query scope so that optimizing queries. Indexing services are support by both programs. Information about indexes in these two DBMS has been extracted in table format shown below:

**Table. 5. Indexes Comparision /82/ /83/ /84/ /85/ /86/ /87/ /88/**

Index Type	MySQL	PostgreSQL
Hash indexes	InnoDB, NDB and MEMORY engines support Hash indexes	Yes
B-tree indexes	PRIMARY KEY, UNIQUE, INDEX, and FULLTEXT)	Needed REINDEX after a database crash
GiST indexes	No	Yes
GIN indexes	No	Yes
Multicolumn Indexes	Limits to 16 columns per index.	PostgreSQL is limited to 32 columns per index.
Multiple Indexes	MySQL supports multiple indexes per table and can use one for each alias of a table; since 5.0 it will also use index merge to use multiple indexes for a single alias.	PostgreSQL supports multiple indexes per query.
Unique Indexes	Yes	Yes, Only B-tree
Expression Indexes	No	PostgreSQL allows you to create indexes based on expressions (which may include calls to immutable functions). This is very handy in case there is a table with relatively stable data (not a lot of inserts / updates) and will often be running a query which involves an expensive calculation - the expression itself can be indexed thus eliminating the need of computing it at query runtime.
Partial Indexes	No	PostgreSQL supports partial indexes: A partial index is an index built over a subset of a table; the subset is defined by a conditional expression (called the predicate of the partial index). The index contains entries for only those table rows that satisfy the predicate. Partial indexes are a specialized feature, but there are several situations in which they are useful. /82/ One major reason for using a partial index is to avoid indexing common values. Since a query searching for a common value (one that accounts for more than a few percent of all the table rows) will not use the index anyway, there is no point in keeping those rows in the index at all. This

		reduces the size of the index, which will speed up queries that do use the index. It will also speed up many table update operations because the index does not need to be updated in all cases.
Full-Text Indexes	Full-text indexes can be used only with MyISAM tables.	PostgreSQL support full text search with index.
Prefix Indexes	MySQL supports prefix indexes. Prefix indexes cover the first $N$ characters of a string column, making the index much smaller than one that covers the entire width of the column, yet still provide good performance characteristics.	With PostgreSQL, prefix indexes are a particular case of Expression Indexes.

#### 4.5. Transaction and concurrency control

Both MySQL and PostgreSQL support transaction function. However, the history that PostgreSQL equipped this function is longer than MySQL. PostgreSQL also supports better transaction function than MySQL.

To illustrate more precisely, MySQL server is only able to provide transaction service with the InnoDB transactional storage engine, since it provides full ACID compliance. This function was added to MySQL version 3.23-max and all versions 4.0 and above. Statements such as SET auto-commit, START TRANSACTION, COMMIT, ROLLBACK and ROLLBACK syntax are used in MySQL local transactions. A dissimilar operation method is used by those non-transactional storage engines in MySQL Server (such as MyISAM) in order to keep the data integrity, the method is named “atomic operations.”/89/ /90/ /91/ /92/ /93/

The Transaction-safe tables (TSTs) in transactional storage engine have advantages/89/ /90/ /91/ /92/ /93/:

- They are safer and easier recovered.
- The COMMIT statement can help to unite numerous statements as well as approved them. (if auto-commit is disabled)
- Changes can be disregard if executing ROLLBACK statement. (If auto-commit is disabled)
- When encounter fails in update process, all changes would be reverted while all changes will alter permanently in nontransaction-safe tables.
- Transaction-safe storage engines performance better when many updates operations along with reads operations concurrently.

Consequently without the transaction service, nontransaction-safe tables (NTSTs) also act well in several aspects, because of /89/ /90/ /91/ /92/ /93/:

- Better speed
- Lower requirements for disk space
- Less space requirements for memory for updates.

There is a possibility to combine the TSTs with NTSTs together to take full use of both of their advantages. However, it is not allowed to mix different storage engines with a transaction, especially when the auto-commit is disabled. /89/ /90/ /91/ /92/ /93/

PostgreSQL supports transaction service for a long history and it is always keep up with latest developments of transaction and attempt to offer all kinds of transactions. Usually, a transaction in PostgreSQL is set up by SQL commands, such as BEGIN and COMMIT commands. The ROLLBACK command is also used in PostgreSQL transactions. In the situation like a commit is not right to submit in the middle-way of a transaction, to command ROLLBACK instead of COMMIT can cancel all updates up to the moment. /89/ /90/ /91/ /92/ /93/

In the fact of that every SQL statement in PostgreSQL is usually executed as a transaction by default. Even though no BEGIN command added to the statement, every individual statement will have a default BEGIN and wrapped with COMMIT if successfully. The “savepoints” is used to control every statements of a transaction in more granularly, since it has the function that select the discard parts and commit the rest. /89/ /90/ /91/ /92/ /93/

Concurrency control is also supported of both MySQL and PostgreSQL. In MySQL, row-level locking is used for InnoDB tables and table-level locking is used for other tables like MyISAM, MEMORY, and MERGE tables. InnoDB storage engine supports four transaction isolation levels, that are READ\_COMMITTED, READ\_UNCOMMITTED, REPEATABLE\_READ and SERIALIZABLE. The SET TRANSACTION ISOLATION LEVEL statement is used to set the isolation level in MySQL. /89/ /90/ /91/ /92/ /93/

PostgreSQL also supports row-level locking as well as table-level locking. Additionally, it supports four levels of transaction isolation, which is also same as MySQL. Even though it is allowed to require any of the four standard transaction isolation levels, internally only two distinct isolation levels are actually available, which are the levels: Read Committed and Serializable. The Read Committed Isolation Level is the default isolation level of PostgreSQL. The command SET TRANSACTION is used to define the transaction isolation level of a transaction in PostgreSQL. Compared with MySQL, PostgreSQL had advisory locks for acquiring locks that are not tightly linked with a single transaction. /89/ /90/ /91/ /92/ /93/

In the situation that multiple users are interested in the same data in the database, a famous approach is created to deal with this situation in DBMS, which is called Multiversion Concurrency Control (MVCC). Though it is not specifically used by PostgreSQL, it commonly exists in many commercial edition DBMSs, like Oracle and so on. PostgreSQL recommend users to MVCC in certain cases because MVCC performances better to meet the users' need. Users can decide when to use it by MVCC's characteristics. The main

characteristics of MVCC are that when executing READ actions, it will not lock the data so that data can be shared with other actions and won't be blocked for other uses; when executing WRITE action, it will create locks but data can still be shared for READ-only actions, the other writing actions should waiting. Because the MVCC service is only available in InnoDB storage engine in MySQL, it limits MySQL's usability in some extent. /89/ /90/ /91/ /92/ /93/

The characteristic of the MVCC model of concurrency control is that reading actions and writing actions needn't to block each other because of the reason that there are no conflict between acquiring reading and writing. In other word, it ensures that fewer clients will be influenced when their needed sources are blocked and locked by other action. However, the MVCC also has the drawbacks. One of the main drawbacks is that it is always essential to continuously clean up those unnecessary data in order to control the occupied space of the disk for programs. /89/ /90/ /91/ /92/ /93/

#### 4.6. Data storage and partition

In fact, the SQL syntax does not have much relationship with the limitation of the physical data storage. The SQL language is able to work with different data carriers independently. The data carriers can be any data structures or media that is able to carry the schemas, tables, rows, or columns to work on it. Though the pre-condition sets no limitation for DBMSs theoretically, the thought that want to ascertain the most proper approaches to store specific data pieces or files is always a fundamental work for all DBMS developers because their eager-mind keeps attempting to promote DBMS better and better. Different file systems and hardware storages or even combination of both have been considered thousands of times. And the physical limitations really exist. /94/ /95/ /96/ /97/ /98/

In the first place, the physical storage limits when working with MySQL is determined by the storage engine types/99/:

- MyISAM: 256 TB
- Memory: RAM
- InnoDB: 64 TB
- Archive: None
- NDB: 384 EB.

The other engines have different storage standards since different designed architectures. The FEDERATED storage engine deals with data from a remote MySQL database and the local database does not offer services like replication or cluster. Consequently, there is no limitation for the FEDERATED storage engine. No data will be stored in local tables because the queries will executed automatically on the remote side with physical storage./97/ /98/ /99/

What's more, there exists a file size limitation, which is also determined by the operation system but not MySQL internally. Otherwise, an error message will occur. Currently, the up-to-date limitation is listed as the table.6 shown below:



**Table. 6. MySQL limits corresponding to operation systems/100/**

Operating System	File-size Limit
Win32 w/ FAT/FAT32	2GB/4GB
Win32 w/ NTFS	2TB (possibly larger)
Linux 2.2-Intel 32-bit	2GB (LFS: 4GB)
Linux 2.4+	(using ext3 file system) 4TB
Solaris 9/10	16TB
MacOS X w/ HFS+	2TB
NetWare w/NSS file system	8TB

There are some commands on the manual book. For windows users, it is more suitable to choose NTFS than FAT and VFAT (FAT32) for the procedure operations. /100/

Moreover, for On Linux 2.2 users, it is possible to that MyISAM table size is over 2 GB if the Large File Support (LFS) patch is applied to the ext2 file system. Large file support (LFS) is used to offer the ability to change the file size limitation larger than 2 GiB on 32-bit operating systems. This patch is included on most Linux distributions which are based on kernel 2.4 or higher. Other patches can also offer help. For instance, patches for ReiserFS can help support file up to 2 TB; with the help of JFS and XFS, petabyte and even larger files is not a mission impossible anymore. /101/

As a DBMS which contains an integrated database server with only a single storage engine, PostgreSQL's physical storage limitation has nothing to do with the engine types. However, other limits existed and can be concluded as below Table.7 shown: /102/

**Table. 7. PostgreSQL limits value/102/**

Limits	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

Analyzing table 6 and table 7, it is obviously that PostgreSQL has the maximum database size and MySQL is the winner of maximum table size. /100/ /102/

MySQL only supports four types of horizontal partitioning. So far there isn't any developing plan about importing vertical partitioning function till MySQL 5.5. /97/ /98/ /99/

- RANGE partitioning. This type of partitioning assigns rows to partitions based on column values falling within a given range. There is an extension for RANGE partitioning named RANGE COLUMNS partitioning.
- LIST partitioning. Similar to partitioning by RANGE, except that the partition is selected based on columns matching one of a set of discrete values. There is an extension for LIST partitioning named LIST COLUMNS partitioning.
- HASH partitioning. With this type of partitioning, a partition is selected based on the value returned by a user-defined expression that operates on column values of rows to be inserted into the table. The function may consist of any expression valid in MySQL that yields a nonnegative integer value. There is an extension for HASH partitioning named HASH Partitioning.
- KEY partitioning. This type of partitioning is similar to partitioning by HASH, except that only one or more columns to be evaluated are supplied, and the MySQL server provides its own hashing function. These columns can contain other than integer values, since the hashing function supplied by MySQL guarantees an integer result regardless of the column data type. There is an extension for KEY partitioning named LINEAR KEY.

MySQL supports subpartitioning - also known as composite partitioning. In MySQL 5.5, it is possible to subpartition tables that are partitioned by RANGE or LIST. Subpartitions may use either HASH or KEY partitioning. MySQL supports a total of 1024 partitions plus subpartitions per table; the partitioning of MySQL, however, cannot be applied to the MERGE, CSV, or FEDERATED storage engines according to the engines' features. /97/ /98/ /99/

PostgreSQL, on the other hand, does not perform more outstanding in partition sector. It supports RANGE and LIST partitioning. HASH partitioning can be obtained from IMMUTABLE functions. The characteristic of PostgreSQL partitioning is that table is able to inherit from a master table. /94/ /95/ /96/

#### **4.7. High availability, replication and recovery**

Generally, the availability reflects the ability of DBMS to dealing with different situations such as crashes or failures and need to restoring data. The sudden accidents are not only the failures that occur on server side but also include failures that happened on operation systems or hardware as well. Therefore, the replication service can be associated with availability easily during analysis. Both MySQL and PostgreSQL offer different solutions to maintain their high availability. /103/ /105/

### 4.7.1. MySQL

MySQL provides two core approaches to maintain its high availability, which are MySQL Replication and MySQL Cluster. Additionally, the third-party solutions of those replication tools can be considered as certain, such as DRBD (Distributed Replicated Block Device) and Heartbeat and so on. /103/ /104/ /105/

Replication service in MySQL makes it possible to replicate data from a MySQL mother database server, also named the master, to single or multiple child database servers, which named the slaves. Because asynchronous replication is setting as default, replicating updates do not need to have a permanent connection to the master. This is crucially important to situations like long-distanced servers connect via a temporary or intermittent network. Moreover, the scope of replicated database can be selected with replication configuration. Even the needed part might be several tables of a database it can also be assigned to b replicated. /103/ /104/ /105/

MySQL supports two basic replication formats, which are Statement Based Replication (SBR) and Row Based Replication (RBR). In addition, it offers a third choice - Mixed Based Replication (MBR). Theoretically, the process of replication is that the master translates the selected content to binary log then passes the log to the slaves to read and store. The SBR is used to copy the whole changed SQL statements to the standard statement-based binary logging format then pass to the slaves. The SBR is like to infect the changes from the master to the slavers. The RBR is used only to write the binary log as event to claim the changes in a single row in tables then apply the changes to the slavers. MBR combines the first two formats together. Usually MBR has the statement-based logging as default setting, and in specific situations the row-based logging will be set automatically. /103/ /104/ /105/

It is notable that statements are not safe with SBR in InnoDB. In replication process, the INSERT statement which contains AUTO\_INCREMENT will block other unrelated INSERT statements in InnoDB storage engine. /103/ /104/ /105/

MySQL 5.5 starts to support semi-synchronous replication as an essential part of MySQL replication. The purpose of this service is to prevent the situation that if the master crashes. The semi-synchronous replication ensures a better data integrity and take advanced of slavers side' work to help the master recovery. The master will not confirm returning a COMMIT before receiving any log receipt from any slaves. /103/ /104/ /105/

There are also several solutions offered by MySQL due to different purposes and situations, such as for backups, scale-out, different master and slave storage engines and so on. The mysqldump can be used to back up a slave, while for back up the raw data from a slave, server should be shut down and copy the file directly. /103/ /104/ /105/

MySQL Cluster, the other solution of availability, is used to share database information among multiple MySQL instances. Dissimilar as other replication, read or write actions of

data in a cluster is accessible to any other node if it has a cluster. The changed information would be distributed to the rest nodes. However, the MySQL Cluster replication requirements are also higher than other replication as it needs a stable LAN connection and geographically separate nodes are not allowed. Additionally, the supported platforms scope is not included maturely number of platforms enough. Even though MySQL Cluster has such limitations, it can perform well in the situations like telecoms and banking. Equally or slightly higher write actions than read actions is the preferred situation by MySQL cluster solution. /103/ /104/ /105/

MySQL supports recovery services by using backup services or directory recovery. There are different kinds of backups, such as logical and physical (raw) backups, online and offline backups, local and remote backups, and snapshot backups as well as full and incremental backups. There are also full and point-in-time recoveries which are offered for the recovery function. Additionally, there is crash recovery for MyISAM tables and transaction recovery is also available. Functions of hot backups and cold backups are available for InnoDB tables in InnoDB and MyISAM tables in MySQL Enterprise Backup. Usually the third-party packages are needed to obtain the hot backups function. Point-in-time recovery can be used by pointing out event time or event positions in MySQL. /103/ /104/ /105/

**Table. 8. Availability under different requirements/105/**

Requirements	MySQL Replication	MySQL Replication + Heartbeat	MySQL Heartbeat + DRBD	MySQL Cluster
Availability				
Automated IP failover	No	Yes	Yes	No
Automated database failover	No	No	Yes	Yes
Typical failover time	User/script-dependent	Varies	Less than 30 seconds	Less than 3 seconds
Automatic resynchronization of data	No	No	Yes	Yes
Geographic redundancy support	Yes	Yes	Yes, when combined with MySQL Replication	Yes, when combined with MySQL Replication

The DRBD (Distributed Replicated Block Device) is a solution that helps replicating data from master to slaves which can be used only on Linux. However Heartbeat cannot be regarded as a replication tool because Heartbeat works only like a server monitor to help MySQL servers to work. Similarly to DRBD, Heartbeat is a Linux running program. Additionally Heartbeat cannot work along. Usually it works to cooperate with MySQL

Replication or DRBD. According to table.8 above, MySQL, Heartbeat and DRBD can work more comprehensively than other combinations. /105/

#### 4.7.2. PostgreSQL

In PostgreSQL 9.0, varied asynchronous replication services are provided, such as streaming replication and hot standby. Hot standby service is not provided by MySQL. Dissimilar to the SBR and RBR of MySQL, the way that PostgreSQL deals with replication service is more careful and full-scale. For instance, the record in PostgreSQL's write-ahead log can help to replay all events that have ever happened in the database. In this way, there is impossible to lose even details of the data because the logs are written advanced in any time. Standby servers also play a pivotal role in high availability. PostgreSQL offers Log-Shipping Standby Servers, which have the ability that is similar to warm standby or log shipping. Hot Standby is a function that helps overcome the fails of the major server. Additionally Hot standby is a new characteristic of PostgreSQL because this service was just provided by some proprietary DBMSs recently. Hot Standby ensures the server keeping opening status and provides read-only queries accessible even if the server is under recovery or standby status. /106/

The work situations of different PostgreSQL replication solutions are offered by the official manual reference. It is easier to compare solutions' strengths and weaknesses in Table.9 which is shown below:

**Table. 9. Comparison of PostgreSQL replication solutions/107/**

Feature	Shared Disk Failover	File System Replication	Hot/Warm Standby Using PITR	Trigger-Based Master-Standby Replication	Statement-Based Replication Middleware	Asynchronous Multi-master Replication	Synchronous Multi-master Replication
Most Common Implementation	NAS	DRBD	PITR	SlonyI	pgpool-II	Bucardo	
Communication Method	Shared disk	Disk blocks	WAL	Table rows	SQL	Table rows	Table rows and row locks
No special hardware required		Yes	Yes	Yes	Yes	Yes	Yes
Allows multiple master servers					Yes	Yes	Yes
No master server overhead	Yes		Yes		Yes		
No	Yes		Yes	Yes		Yes	

waiting for multiple servers							
Master failure will never lose data	Yes	Yes			Yes		Yes
Standby accept read-only queries			Hot Only	Yes	Yes	Yes	Yes
Per-table granularity				Yes		Yes	Yes
No conflict resolution necessary	Yes	Yes	Yes	Yes			Yes

Data Partitioning and Multiple-Server Parallel Query Execution are also recommended by PostgreSQL global developing group. Information of several recommend replication packages is provided by PostgreSQL as well, including PGCluster, DBBalancer, PostgreSQL table comparator, SkyTools, Sequoia, Mammoth Replicator, Cybercluster, rubyrep (asynchronous, master-to-master), bondreplicate (asynchronous, master-to-master) and so on. Here will introduce two main replication tools of PostgreSQL – Slony-I and PGCluster. /107/

### Slony-I replication

One of the most common PostgreSQL replication tool is called Slony-I, which was design and build by a group of developing members from PostgreSQL community. The developing group of Slony-I is leading by a member named is Jan Weick from the PostgreSQL core team. Slony-I is a master to multiple-slaves replication system. With Slony-I's well performance and the success of Slony-I, the belief that the third-party packages or the plug-in tools of PostgreSQL are not reliable was rebutted. /107/ /108/

However Slony-I contains some weaknesses as well. Usually, SlonyI's performance might be regarded as slow. Additionally Slony-I requires more resources than those built-in replication services because built-in replication services use different approaches to obtain replication. Slony-I uses SQL and triggers but the built-in replication, for instance in MySQL, uses binary logs. These weaknesses do make Slony-I not suitable for the situations that require high performance and high intensity deploying. /107/ /108/

There are numerous reasons why Slony-I has weaknesses. First of all, the main reason is the replication record carriers. Because SQL statements and triggers are used in Slony-I, the other carriers like binary logs are easier to read. /107/ /108/

Secondly, the communication costs in Slony-I grow in a quadratic fashion, and the variable in the function is depend on the number of cluster replication nodes/64/. Consequently Slony-I needs more resource for processing works. This is the fundamental reason that

Slony-I does not suit to work with large number of clusters. The conserve formula can be used to illustrate how the situations are changed when replication process is work with Slony-I to compare the situations of MySQL in the same condition. It is obvious that the cost of Slony-I grows higher and higher than MySQL's when the servers number is increasing. /107/ /108/ /109/

- If there are 2 servers, for MySQL: 2 while for PostgreSQL:  $2*(2^2) = 8$ .
- If there are 4 servers, for MySQL: 4 while for PostgreSQL:  $2*(4^2) = 32$ .
- If there are 6 servers, for MySQL: 6 while for PostgreSQL:  $2*(6^2) = 72$ .

### **PGCluster**

PGCluster is another well-known replication tool based on PostgreSQL. PGCluster is used to dealing with the synchronous replication of multiple master servers. PGCluster has two functions, one is load-sharing, and the other one is high-availability and it is not suitable for the work that requests high-performance and a lot of write actions. Because PGCluster's working flow in the synchronous replication system needs the pre-condition that is finishing all current write actions in the servers first. Therefore, this characteristic makes it performance well in the condition with few write actions. /107/ /110/

PostgreSQL supports recovery with backups function or without backups function. PostgreSQL supports all kinds of recoveries that MySQL has. Additionally, PostgreSQL supports hot and cold recovery without the MySQL-kind limitation. /107/ /110/

## **4.8. Encryption and authentication**

Both MySQL and PostgreSQL support the identity verification services and offer encryption techniques. However PostgreSQL support this part better than MySQL.

MySQL support database level identity verification, which is based on the format of password verification. Data can be encrypted by user-defined password. Functions like AES\_ENCRYPT and AES\_DECRYPT can encrypt or decrypt specific columns. Additionally, SSL and SSH can also be used to encrypting a network. The hashing mechanism in MySQL is used to encrypting the password to enhance the security. Even though neither of LDAP nor RBAC is provided by MySQL, pluggable authentication and proxy users are newly supported by MySQL authentication. The plug-ins can be created to use external authentication methods such as LDAP, Kerberos, PAM, or Windows login IDs in MySQL 5.5.7. Secure Shell or SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices. Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communications security over the Internet. The Lightweight Directory Access Protocol (LDAP) is an application protocol for reading and editing directories over an IP network. Role-based access control (RBAC) is an approach to

restricting system access to authorized users in computer systems security. /55/ /111/ /112/ /113/ /114/ /115//116/

PostgreSQL performance more attractive in this part since it offers more choices than MySQL. PostgreSQL, however, provide rich approaches of authentication, such as Trust authentication, Password authentication, GSSAPI authentication, SSPI authentication, Kerberos authentication , Ident-based authentication, RADIUS authentication, Certificate authentication as well as LDAP authentication and PAM authentication, which mostly do not supported by MySQLitself. Similar to MySQL, PostgreSQL uses SSL to encrypt networks; while it uses contrib function library pgcrypto to encrypt data in specific column, data will be decrypt on server then pass to the client. Commonly, the user password in PostgreSQL will be saved as MD5 hashes to improve the security level. /55/ /111/ /112/ /113/ /114/ /115//116/

#### **4.9. Support for distribution and parallel processing**

MySQL Server can work in client/server or embedded systems. The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). MySQL Server is also provided as an embedded multi-threaded library that is used to connecting applications to form a smaller, faster, easier-to-manage standalone product./117/



## 5. CONCLUSIONS

According to the research, PostgreSQL definitely has the all-round functions as a DBMS. MySQL is easier to operate and has a higher reputation and influence than PostgreSQL among public, which can somehow make up for the shortages in functions. The higher reputation also makes people believe that MySQL do have the potential ability to improve to a higher level.

With the experiences of deeply researching MySQL and PostgreSQL, I have obtained a lot. First of all, database management system is no longer a mystery to me in the reason that I understand the developing history and concept of it. Secondly, I not only acquired information thorough understanding every aspects of MySQL and PostgreSQL, but also knowledge of Linux-like and Unix-like operating systems as well; this will crucially helpful for my further learning definitely. The last but not the least, this is the first time I do a project totally on my own, which turns out to be the unique experience to inspire me a lot to be more brave and independent.

The research is attractive to me because I have a strong interest in information system, which is linked tightly with DBMS. According to the research experience, I have the opportunities enable myself to explore my desirable field as well. Within the help of this research, I have established the fundamental basic knowledge of DBMS and obtained more confident about my future study plan. Additionally, the self-learning process formed me a more clear self-awareness because I proved my professional knowledge and study ability.

Transaction, advanced queries, query optimization as well as concurrency, they are the key points in my research. The aim - covered audiences are not only those new beginners in DBMS but also professional individuals. After experienced continuous acquisition, MySQL has drawn a great attention and PostgreSQL is increasing shining as an open source DBMS, the thesis will follow the main stream of public interests to show the sparkle from the impact of MySQL's and PostgreSQL's culture and techniques.

## 6. REFERENCES

### List of the sources:

/1/ Jiaquan Shi, Database Management Theory, 1<sup>st</sup> Edition, Chapter 1, p. 1-4, Tsinghua University Press, August 2007.

/2/ Peixi Zhao, Computer Basis, 1<sup>st</sup> Edition, Chapter 7, p. 221-230, China Railway Publishing House, September 2005.

/3/ C J Date, An Introduction to Database Systems – Instructor's Manual, 8<sup>th</sup> Edition, Chapter 1-4, p. 13-57, 2003.

/4/ Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems 4th Edition, CHA'IJTER 1 Databases and Database Users, p. 3-24, Addison-Wesley, 2004.

/5/ Bob DuCharme, Bob DuCharme's weblog, [WWW-document], <<http://www.snee.com/bobdc.blog/2005/12/25-years-of-database-history-s-1.html>>, 20.02.2011.

/6/ Steven P. Ross, Ezines Articles, [WWW-document], <<http://ezinearticles.com/?The-History-of-Database-Management-Systems&id=904394>>, 20.02.2011.

/7/ zaadmin, SybaseTeam, [WWW-document], <<http://www.sybaseteam.com/what-is-database-management-system-dbms-t-382.html>>, 20.02.2011.

/8/ Joan C. Nordbotten, Multimedia Information Retrieval Systems, 2nd Edition, Dept. of Information and Media Science, University of Bergen, Norway, Chapter 2 Management Systems for Media Databases, December 2008.

/9/ ReferenceAnswer, [WWW-document], <<http://www.answers.com/topic/database-management-system>>, 04.03.2011.

/10/ Skyteam, [WWW-document], <<http://www.sybaseteam.com/history-of-dbms-t-383.html>>, 07.08.2009.

/11/ Arun Kumar Ramesh, Eggheadcafe, [WWW-document], <<http://www.eggheadcafe.com/community/aspnet/2/10182852/sql.aspx>>, 20.03.2011.

/12/ Web Vitality Ltd, [WWW-document], <<http://sql-tutorials.com/>>, 20.03.2011.

/13/ Troels , Comparison of different SQL implementations, [WWW-document], <<http://troels.arvin.dk/db/rdbms/#legend>>, 20.03.2011.

- /14/ Oracle, Oracle® Database SQL Language Reference 11g Release 1 (11.1), [WWW-document],  
<[http://download.oracle.com/docs/cd/B28359\\_01/server.111/b28286/intro002.htm](http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/intro002.htm)>,  
20.03.2011.
- /15/ Whitemarsh Information Systems Corporation, [WWW-document],  
<<http://www.wiscorp.com/SQLStandards.html>>, 20.03.2011.
- /16/ Randy J. Jarger, Managing and Using MySQL Simplified Chinese Edition, 2nd Edition, Chapter 1, p. 1-20, O'Reilly, April 2002.
- /17/ MySQL, [WWW-document], <<http://dev.mysql.com/doc/refman/5.5/en/history.html>>,  
04.03.2011
- /18/ MySQL, [WWW-document], <<http://dev.mysql.com/doc/refman/5.5/en/what-is-mysql.html>>, 04.03.2011
- /19/ MySQL, [WWW-document], <<http://www.mysql.com/products/>>, 05.03.2011
- /20/ Ian Gilfillan, Mastering MySQL 4 Simplified Chinese Edition, 1<sup>st</sup> Edition, Chapter 1, p. i-34, Sybex, January 21, 2003.
- /21/ Baron Schwartz, Peter Zaitsev and so on, High Performance MySQL Simplified Chinese Edition, 2<sup>nd</sup> Edition, Preface, O'Reilly, 2008.
- /22/ Timeline results for mysql history,  
[WWW-document], <[http://www.google.com.hk/search?q=mysql+history&hl=zh-CN&safe=strict&client=aff-maxthon&hs=QrZ&channel=channel4&source=hp&biw=1236&bih=587&prmd=ivns&tbs=tl:1&tbo=u&ei=7N6RTe6JLMfbsgbUu5jQBg&sa=X&oi=timeline\\_result&ct=title&resnum=11&ved=0CHUQ5wIwCg](http://www.google.com.hk/search?q=mysql+history&hl=zh-CN&safe=strict&client=aff-maxthon&hs=QrZ&channel=channel4&source=hp&biw=1236&bih=587&prmd=ivns&tbs=tl:1&tbo=u&ei=7N6RTe6JLMfbsgbUu5jQBg&sa=X&oi=timeline_result&ct=title&resnum=11&ved=0CHUQ5wIwCg)>, 04.03.2011.
- /23/ PostgreSQL, [WWW-document], <<http://www.postgresql.org/docs/9.0/static/intro-what-is.html>>, 05.03.2011.
- /24/ Command prompt, Inc, Introduction to Relational Databases, [WWW-document],  
<<http://www.commandprompt.com/ppbook/x1277>>, 05.03.2011.
- /25/ PostgreSQL, [WWW-document],  
<<http://www.postgresql.org/docs/9.0/static/history.html>>, 05.03.2011
- /26/ Gregory Smith, PostgreSQL 9.0 High performance, 1<sup>st</sup> Edition, Chapter 1, p. 7-12, Packt Publishing, October 2010.
- /27/ Simon Riggs and Hannu Krosing, PostgreSQL 9 Administration Cookbook, 1<sup>st</sup> Edition, Preface and Chapter 1, p. 1-7, Packt Publishing, October 2010.

/28/ Bruce Momjian, PostgreSQL Introduction and Concepts, 1<sup>st</sup> Edition, Chapter 1, p. 1-4, ADDISON-WESLEY, 2001.

/29/ PostgreSQL, [WWW-document],  
<<http://www.postgresql.org/docs/9.0/static/release.html>>, 05.03.2011

/30/ Action Technology, [WWW-document],  
<[http://www.actionsky.com/why\\_mysql/market\\_share.jsp](http://www.actionsky.com/why_mysql/market_share.jsp)>, 05.03.2011.

/31/ IT168, [WWW-document],  
<[http://tech.it168.com/a2010/0107/834/000000834281\\_all.shtml](http://tech.it168.com/a2010/0107/834/000000834281_all.shtml)>, 06.03.2011

/32/ MySQL, MySQL 5.5 Reference Manual, Chapter 2 Installation and upgrade MySQL, p. 34-105

/33/ winksir, Ubuntu, [WWW-document],  
<<http://forum.ubuntu.org.cn/viewtopic.php?f=44&t=319886&start=0>>, 08.03.2011.

/34/ FreeBSD, [WWW-document], <<http://www.freebsd.org/where.html>>, 08.03.2011.

/35/ PCBSD, [WWW-document], <<http://www.pcbsd.org/>>, 08.03.2011.

/36/ KDE, [WWW-document], <<http://www.kde.org/>>, 08.03.2011.

/37/ JianXinTongTou, ChinaUnix.net, [WWW-document],  
<<http://www.chinaunix.net/jh/5/591045.html>>, 08.05.2011.

/38/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 15 Installation from source code, p. 341- 365

39/ PostgreSQL, [WWW-document], <<http://www.postgresql.org/docs/9.0/static/install-short.html>>, 10.03.2011.

/40/ JasonK, JasonK's Blog, [WWW-document],  
<<http://jasonk2600.wordpress.com/2010/01/11/installing-postgresql-on-freebsd/>>, 08.05.2011.

/41/ MySQL, MySQL 5.5 Reference Manual, Chapter 1, p. 1-5

/42/ Joytip, JoyTip the Life, [WWW-document],  
<<http://hi.baidu.com/joytip/blog/item/45f0a40858e41fd462d986c5.html>>, 10.03.2011.

/43/ PostgreSQL, PostgreSQL Reference Manual 9.0, Preface.

/44/ Web Hosting Consultant, [WWW-document],  
<<http://www.mybestratedwebhosting.com/best-web-hosting-tips/mysql-vs-postgresql.html>>, 11.03.2011.

/45/ Ezine Articles, [WWW-document], <<http://ezinearticles.com/?Postgre-SQL-Vs-MySQL&id=2264294>>, 11.03.2011.

/46/ Steven C Jackson, PostgreSQL and MySQL: Comparing Licensing, Functionality, and Reliability, [WWW-document], <<http://www.scribd.com/doc/23956013/PostgreSQL-and-MySQL-Comparing-Licensing-Functionality-and-Reliability>>, 11.03.2011.

/47/ DBMS2, [WWW-document], <<http://www.dbms2.com/2008/07/07/postgresql-vs-mysql-as-per-enterprisedb/>>, 11.03.2011.

/48/ BitsCN, [WWW-document], <<http://www.bitscn.com/pdb/mysql/200709/114175.html>>, 11.03.2011.

/49/ Jim Mlodgenski, [WWW-document], <<http://www.eweek.com/c/a/Linux-and-Open-Source/PostgreSQL-vs-MySQL-How-to-Select-the-Right-OpenSource-Database/>>, 11.03.2011.

/50/ Cygwin, [WWW-document], <<http://www.cygwin.com/>>, 12.03.2011.

/51/ MySQL, MySQL 5.5 Reference Manual, Chapter 13.6 The InnoDB Storage Engine, p. 1106

/52/ Robert Haas, EnterpriseDB, [WWW-document], <<http://blogs.enterprisedb.com/2011/02/01/mysql-vs-postgresql-part-2-vacuum-vs-purge/>>, 12.03.2011.

/53/ MySQL, MySQL 5.5: Storage Engine Performance Benchmark for MyISAM and InnoDB, p. 10.

/54/ hlbng, JavaEye, [WWW-document], <<http://hlbng.javaeye.com/blog/392704>>, 12.03.2011.

/55/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 19.3. Authentication methods, p. 442-449

/56/ PostGIS, [WWW-document], <<http://postgis.refrations.net/>>, 12.03.2011.

/57/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 38-39. Procedural languages, p. 827-844

/58/ MySQL, MySQL 5.5 Reference Manual, Chapter 1.5.1. Scalability Improvements, p. 10

/59/ WiseGeek, [WWW-document], <<http://www.wisegeek.com/what-is-dbms.htm>>, 11.03.2011.

- /60/ DBmaker, [WWW-document],  
<[http://www.dbmaker.com.tw/reference/manuals/tutorial/tutorial\\_03.html](http://www.dbmaker.com.tw/reference/manuals/tutorial/tutorial_03.html)>, 07.03.2011
- /61/ MySQL, [WWW-document], <http://dev.mysql.com/doc/refman/5.5/en/pluggable-storage-overview.html>> ,
- /62/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 8 Data Type. p. 100-147
- /63/ MySQL, MySQL 5.5 Reference Manual, Chapter 10 Data Type, p. 739-765
- /64/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 5 Data Definition, p. 47-76
- /65/ MySQL, MySQL 5.5 Reference Manual, Chapter 10.1.1. Overview of Numeric Types, p. 745-746
- /66/ MySQL, MySQL 5.5 Reference Manual, Chapter 12.2.5. INSERT Syntax, p. 954-959
- /67/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 37.3. Rules on INSERT, UPDATE, and DELETE, p. 811-816
- /68/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 35.13.6. MERGES, p. 782
- /69/ MySQL, MySQL 5.5 Reference Manual, Chapter 1.8.6. How MySQL Deals with Constraints, p. 24-26
- /70/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 4. SQL Syntax p. 24-46
- /71/ MySQL, MySQL 5.5 Reference Manual, Chapter 1.8. MySQL Standards Compliance, p. 17-27
- /72/ MySQL, MySQL 5.5 Reference Manual, Chapter 19. Stored Programs and Views, p. 1810-1826
- /73/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 35.7. Procedural Language Functions, p. 747
- /74/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 38-39, p. 827-844
- /75/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 36. Triggers, p. 794-803
- /76/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 9.25. Trigger Functions, p. 257
- /77/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 39.9. Trigger Procedures, p. 862
- /78/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 7.2.1.3. Subqueries, p. 86

- /79/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 9.20. Subquery Expressions, p.232-235
- /80/ MySQL, MySQL 5.5 Reference Manual, Chapter 12.2.10. Subquery Syntax, p. 986
- /81/ MySQL, MySQL 5.5 Reference Manual, Chapter E.4. Restrictions on Subqueries, p. 3108
- /82/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 11-12, p. 268-316
- /83/ MySQL, MySQL 5.5 Reference Manual, Chapter 7.3. Optimization and Indexes, p. 574-577
- /84/ MySQL, MySQL 5.5 Reference Manual, Chapter 11.9. Full-Text Search Functions, p. 827-839
- /85/ MySQL, MySQL 5.5 Reference Manual, Chapter 13.9. The MEMORY Storage Engine, p. 1269
- /86/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 11.3. Multicolumn Indexes, p. 329
- /87/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 11.6. Unique Indexes, p. 332
- /88/ MySQL, MySQL 5.5 Reference Manual, Chapter 5.1.1. Server Option and Variable Reference, p. 345
- /89/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 3.4. Transactions, p. 15-17
- /90/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 13. Concurrency Control, p. 316-327
- /91/ MySQL, MySQL 5.5 Reference Manual, Chapter 7.5. Optimizing for InnoDB Tables, p. 583-587
- /92/ MySQL, MySQL 5.5 Reference Manual, Chapter 12.3. MySQL Transactional and Locking Statements, p. 996-1008
- /93/ MySQL, MySQL 5.5 Reference Manual, Chapter 13.1. Comparing Transaction and Nontransaction Engines, p. 1096
- /94/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 54. Database Physical Storage, p. 1594-1602
- /95/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 10.4. Value Storage, p. 264
- /96/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 5.9. Partitioning, p. 69-76

- /97/ MySQL, MySQL 5.5 Reference Manual, Chapter 10.5. Data Type Storage Requirements, p. 761
- /98/ MySQL, MySQL 5.5 Reference Manual, Chapter 18. Partitioning, p. 1676-1808
- /99/ MySQL, MySQL 5.5 Reference Manual, Chapter 13. Storage Engines, p. 1094-1246
- /100/ MySQL, [WWW-document], <<http://dev.mysql.com/doc/refman/5.0/en/full-table.html>>, 13.03.2011
- /101/ Opengroup, [WWW-document], <<http://www.opengroup.org/platform/lfs.html>>, 13.03.2011
- /102/ PostgreSQL, [WWW-document], <<http://www.postgresql.org/about/>>, 15.03.2011
- /103/ MySQL, MySQL 5.5 Reference Manual, Chapter 14. High Availability and Scalability, p. 1249-1357
- /104/ MySQL, MySQL 5.5 Reference Manual, Chapter 17. Replication, p. 1671-1765
- /105/ MySQL, MySQL 5.5 Reference Manual, Chapter 14. High Availability and Scalability, p.1283
- /106/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 25-26, p. 495-512
- /107/ PostgreSQL, PostgreSQL Reference Manual 9.0, Chapter 25.1. Comparison of different solutions, p.556-557
- /108/ Slony-I, [WWW-document], <<http://www.slony.info/>>, 13.03.2011.
- /109/ Linuxfinance, [WWW-document], <<http://linuxfinances.info/info/slonylistenercosts.html>>, 15.03.2011
- /110/ PGCluster, [WWW-document], <<http://pgcluster.projects.postgresql.org/>>, 15.03.2011.
- /111/ Network Working Group of the IETF, The Secure Shell (SSH) Authentication Protocol, January 2006, RFC 4252,
- /112/ T. Dierks, The Transport Layer Security (TLS) Protocol, Version 1.2, RTFM, Inc, August 2008
- /113/ Vassiliki Koutsonikola, Athena Vakali, [WWW-document], CS Digital Library, <<http://www.computer.org/portal/web/csdl/doi?doc=doi/10.1109/MIC.2004.44>>, 15.03.2011
- /114/ Ferraiolo, D.F. and Kuhn, D.R, "Role-Based Access Control", 15th National Computer Security Conference, October 1992, p. 554–563.



/115/ MySQL, MySQL 5.5 Reference Manual, Chapter 11.13 Encryption and Compression Functions, p. 851-855

/116/ MySQL, MySQL 5.5 Reference Manual, D.1.5. Changes in MySQL 5.5.7 (14 October 2010), p. 2865

/117/ MySQL, MySQL 5.5 Reference Manual, Chapter 1.3.1 What is MySQL?, p36

## 7. LIST OF APPENDICES

### APPENDIX 1 For the Table.2

<sup>1</sup> Features only available in Commercial Editions.

<sup>2</sup> Annual Subscription: is defined as the right to use the specified program(s) in accordance with the applicable license metric and to receive Oracle Software Update License & Support for the specified program(s) for the term specified on the order.

<sup>3</sup> Oracle Premier Support for MySQL is included in Annual Subscriptions (no extra cost)

<sup>4</sup> Server: is defined as the computer on which the programs are installed. A Server license allows you to use the licensed program on a single specified computer.

<sup>5</sup> Socket: is defined as a slot that houses a chip (or a multi-chip module), which contains a collection of one or more cores. Regardless of the number of cores, each chip (or multi-chip module) counts as a single socket. All occupied sockets on which the Oracle programs is installed and/or running must be licensed.

Pricing is "per Server", for 2 classes: Servers with 1-4 Sockets, and Servers with 5+ Sockets

## APPENDIX 2 Test Environment and Result.

These tests are made manually during the research for thesis need. They are not that strictly follow the scientific test criteria. However, the aim of these tests is to provide some information and direction for further research. These might help to inspire new ideas.

The environment of testing is:

CPU: Intel Core2 Duo P8400 @ 2.26 GHz

EMS memory: 3.45 GB

Platform: Ubuntu-10.10

Webmin: 1.540

PHP: 5.5

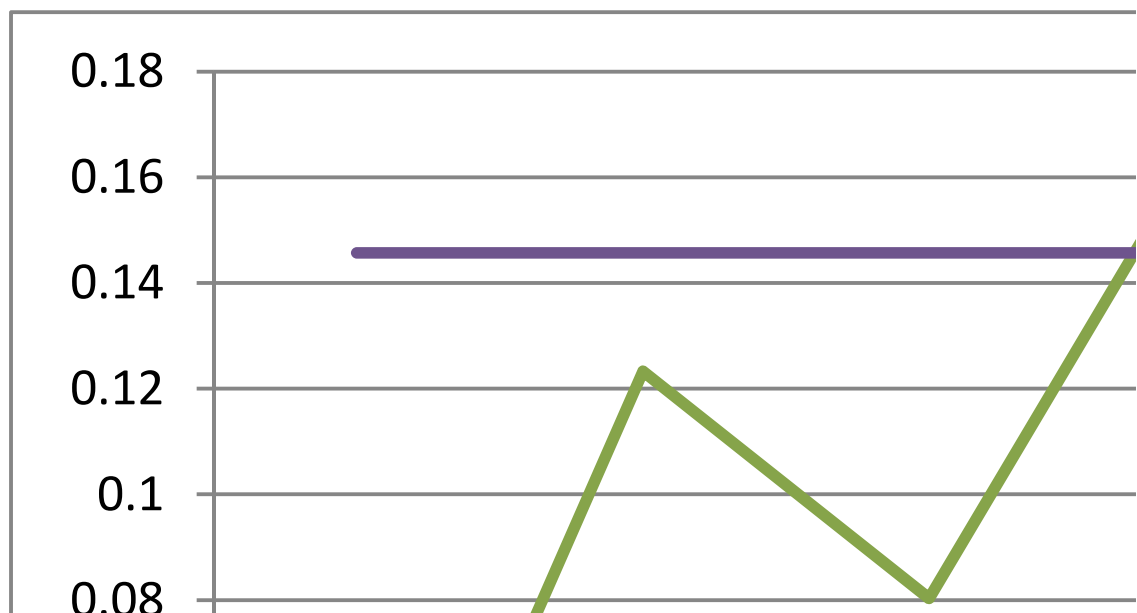
Apache Webserver: 2.2.16

PostgreSQL: 8.4.7 (With schemas)

MySQL: 5.1.49

PhpMyAdmin: 3.3.7

There are two kinds of tests. The first one is to execute certain numbers of actions in PHP pages include INSERT, SELECT UPDATE and DELETE. The spend time was record and replay 5 times every 20 seconds for each storage engines – MyISAM, InnoDB and PostgreSQL. Average values of the time were calculated based on those test results. The second test was accessing the database in 10 times via PHP pages in the same time and recorded the spend time. All the execute codes were not optimized and transaction services or other helped advanced query functions were not used during the testing. The figure below shows in result of the first test.



In fact, InnoDB storage engine and PostgreSQL engine were comparable. They are similar storage engines to the users. The reason that test MyISAM was because it was the previous default storage engine of MySQL. According to the test, PostgreSQL storage engine used less time but more stable. InnoDB storage engine operated the same actions using time

which was more than four times of the PostgreSQL's. In addition, InnoDB's performance was not stable as PostgreSQL's but changed dramatically.

MyISAM is another type of pluggable storage engines of MySQL. MyISAM doesn't implement with many functions and it performed well read-only situations. According to the test result, it performed stable. The unit used in tests is second. The detailed test result is listed as below:

### **MyISAM:**

1. In MyISAM table, altogether : 0.0013349056243896 seconds  
Select times100002  
Insert times999999  
Delete times99997  
Update times100003

2. In MyISAM table, altogether : 0.00098180770874023 seconds  
APPENDIX 2/2

Select times100001  
Insert times1000000  
Delete times99999  
Update times100001

3. In MyISAM table, altogether : 0.0026040077209473 seconds  
Select times100002  
Insert times1000000  
Delete times99996  
Update times100003

4. In MyISAM table, altogether : 0.003093957901001 seconds  
Select times100002  
Insert times1000000  
Delete times99997  
Update times100002

5. In MyISAM table, altogether : 0.0011970996856689 seconds  
Select times100001  
Insert times999999  
Delete times99999  
Update times100002

MyISAM Average Value: 0.001842355728149406 seconds

### **InnoDB:**

1. In InnoDB table, altogether : 0.21641802787781 seconds

Select times100002  
Insert times1000002  
Delete times99996  
Update times100001

2. In InnoDB table, altogether : 0.12331509590149 seconds

Select times100002  
Insert times1000000  
Delete times99999  
Update times100000

3. In InnoDB table, altogether : 0.080280065536499 seconds

Select times100002  
Insert times999999  
Delete times99998  
Update times100002

4. In InnoDB table, altogether: 0.17127799987793 seconds

Select times100000  
Insert times1000001  
Delete times99997  
Update times100003

5. In InnoDB table, altogether : 0.13699793815613 seconds

Select times100003  
Insert times1000000  
Delete times99996  
Update times100002

InnoDB Average Value: 0.1456578254699718 seconds.

### **PostgreSQL:**

1. Altogether : 0.036657094955444 seconds

Select times100001  
Insert times1000001  
Delete times99996  
Update times100003

2. Altogether : 0.027704954147339 seconds

Select times100001  
Insert times999999  
Delete times99998  
Update times100003

3. Altogether : 0.027704954147339 seconds

Select times100001  
Insert times999999

Delete times99998  
Update times100003

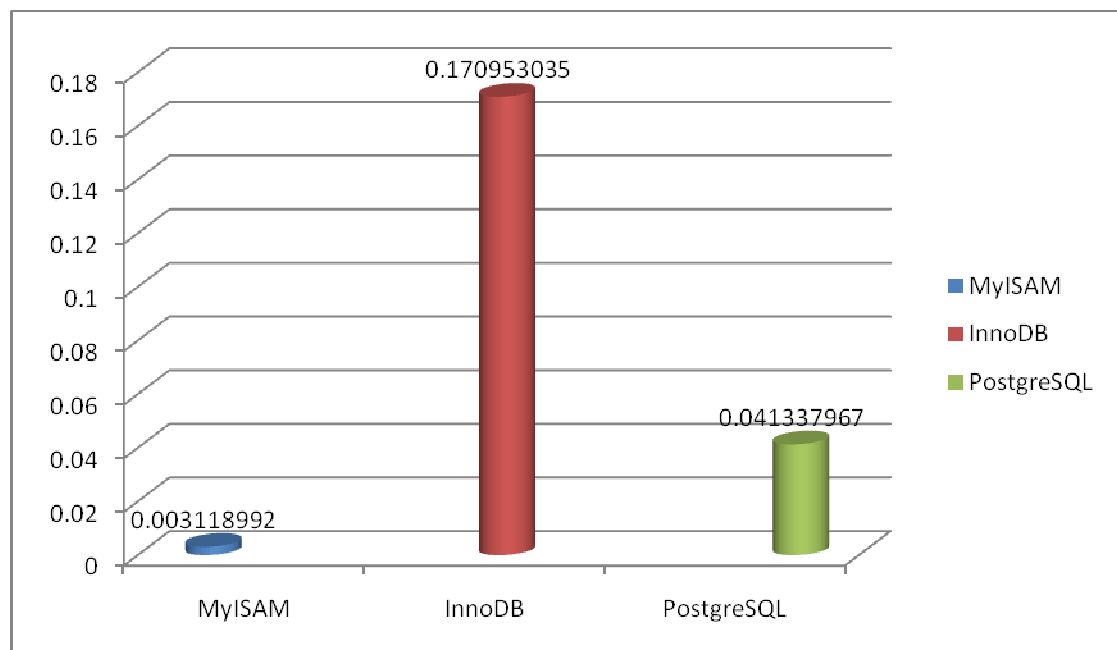
4. Altogether : 0.028504133224487 seconds

Select times99999  
Insert times1000000  
Delete times99998  
Update times100004

5. Altogether : 0.045192956924438 seconds

Select times99999  
Insert times1000001  
Delete times99999  
Update times100002

PostgreSQL Average Value: 0.0331528186798094 seconds.



The above figure shows the test result of the second test. The unit used in this test is also second. According to test result, PostgreSQL storage engine used less time than InnoDB storage engine when 10 read actions happened in the same time. Firefox was the web browser that was used in second test.

The test SQL code and Php code of InnoDB and PostgreSQL of the first test is also listed here for help. The SQL code was used to create table in MySQL. The Php code was used to run specific actions.

The SQL code and php code of InnoDB is:

```

CREATE TABLE IF NOT EXISTS test_innodb (
  id int(10) unsigned NOT NULL AUTO_INCREMENT,
  aid upid int(10) unsigned NOT NULL DEFAULT '0',
  bid mediumint(8) unsigned NOT NULL DEFAULT '0',
  cid mediumint(8) unsigned NOT NULL DEFAULT '0',
  username char(15) NOT NULL DEFAULT '',
  phone int(10) unsigned NOT NULL DEFAULT '0',
  msg text NOT NULL,
  ip char(20) NOT NULL DEFAULT '',
  level smallint(6) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY ( id ),
  KEY bid ( bid )
) ENGINE=InnoDB ;

```

```

<?PHP

header("Content-type: text/html; charset=utf-8");

$in_sql = "INSERT INTO test_innodb (aid ,bid ,cid ,username ,phone ,msg ,ip ,level)VALUES (
'5', '5', '5', 'firstname', '5555555555', 'abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop
abcdefghijklmnopabcdefghijklmnop', '127.0.0.1', '5'
)"; // Insert data directly

$id_in=999997; //insert times
//select data, keep one for later use
$sse_sql="select * from test_innodb LIMIT 1";
//select times
$id_se=99998;
// random delete id
$idde=99;
$del_sql ="DELETE FROM test_innodb WHERE id = ".$idde." ";
//delete times
$id_de=99996;
// random update id
$idup=100;
$up_sql = "UPDATE test_innodb SET bid = '4' WHERE id =".$idup." ";
//update times
$id_up=100000;

//Connect database
$link = mysql_connect('localhost', 'testing', '123456') or die('Could not connect: ' .
mysql_error());
echo 'Connected MySQL successfully';
mysql_select_db('mydb',$link) or die('Could not select database'. mysql_error());

//Start timer
$start_time = microtime_float();
// Among 10000 times for function, operate SQL randomly
for($i=0;$i<10;$i++) {
  $a=rand(1,4);
  switch ($a) {
    case $a==1:
      $i_s++;
      //select
      $result = mysql_query($sse_sql) or die('Query failed: ' . mysql_error());
      while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
        rand(0,1)?$idde=$line['id']:$idup=$line['id'];
      }
      mysql_free_result($result); //show the result
      break;
    case $a==2:
      $id_in++;
      //insert
      $result = mysql_query($in_sql) or die('Query insert failed: ' . pg_last_error());
      break;
    case $a==3:
      $id_de++;
      //delete
      $result = mysql_query($del_sql) or die('Query delete failed: ' . pg_last_error());
      break;
    case $a==4:
      $id_up++;

```

```

        //update
        $result = mysql_query($up_sql) or die('Query update failed: ' . pg_last_error());
        break;
    default:
        echo "Deault";
        break;
    }
}

//End and record time
$end_time = microtime_float();

// close connection
mysql_close($link);

echo "<br />In InnoDB table, altogether : ".$end_time-$start_time)." seconds";
echo "<br /> Select times".$id_se;
echo "<br /> Insert times".$id_in;
echo "<br /> Delete times".$id_de;
echo "<br /> Update times".$id_up;

//from php manual
function microtime_float()
{
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}
?>

```

The php test code of PostgreSQL was similar to InnoDB's. Therefore, only the SQL code of PostgreSQL listed here. The way SQL create table in PostgreSQL was a little bit different from MySQL's.

```

CREATE TABLE test_postgres (
    id serial,
    aid integer NOT NULL DEFAULT '0',
    bid integer NOT NULL DEFAULT '0',
    cid integer NOT NULL DEFAULT '0',
    username char(15) NOT NULL DEFAULT '',
    phone integer NOT NULL DEFAULT '0',
    msg text NOT NULL,
    ip char(20) NOT NULL DEFAULT '',
    level integer NOT NULL DEFAULT '0',
    PRIMARY KEY (id,bid)
) ;

```